Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

На правах рукописи

Волков

Волков Мария Сабина Александровна

ИССЛЕДОВАНИЕ КОМБИНАТОРНЫХ СВОЙСТВ И ОЦЕНКА ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ ЗАДАЧ РЮКЗАЧНОГО ТИПА

Специальность: 1.2.3. «Теоретическая информатика, кибернетика»

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
д.ф.-м.н., профессор
Гордеев Эдуард Николаевич

Москва – 2026

Оглавление

Введение

Актуальность темы исследования. Исследование вычислительной сложности фундаментальных комбинаторных задач занимает центральное место в проблематике теоретической информатики. В данном контексте особый интерес представляет класс NP-полных задач, поскольку его изучение непосредственно связано с проблемой соотношения классов P и NP — одной из ключевых открытых проблем современной математики. Канонические NP-полные задачи, к числу которых относится задача о рюкзаке, служат не только объектом исследования сами по себе, но и инструментом для доказательства NP-полноты других задач посредством сведения. Заложенный в работах С. Кука, Л. Левина и Р. Карпа фундамент теории NP-полноты создал строгий базис для классификации вычислительной сложности, а последующие исследования Д. Кнута, Д. Хопкрофта, М. Гэри, Д. Джонсона, А. Бородина, А. Кобхэма, Д. Эдмондса, М. Рабина, А. Мейера, Л. Стокмейера, Л. Адлемана углубили понимание иерархии классов сложности и структуры вычислимых функций.

Исторически мощным катализатором исследований в этой области выступила криптография. Предложенная Р. Мерклом и М. Хеллманом в 1978 году криптосистема, основанная на задаче о рюкзаке, стала одним из первых практических применений NP-трудности для построения асимметричных шифров. Однако последующая атака А. Шамира, использовавшая методы целочисленного программирования, наглядно продемонстрировала, что сама по себе принадлежность задачи к классу NP-трудных не является гарантией стойкости. Это стимулировало формирование двух взаимосвязанных направлений исследований. С одной стороны, работы Б. Шора, Р. Ривеста, Д. Накааче, Ж. Стерна, Т. Окамото, К. Танаки, С. Учиямы, Я. Мураками, Т. Насако, Б. Ван, В. О. Осипяна и В. В. Подколзина были направлены на конструирование модифицированных и усложненных вариантов рюкзачных криптосистем, устойчивых к известным методам решения. Параллельно, исследования Л.

Адельмана, А. М. Одлыжко, Д. Лагариаса, Х. Ленстры, М. Костера, А. Жу, С. Палита, С. Синхи, М. Моллы, А. Ханры и Д. М. Мурина концентрировались на разработке новых аналитических методов, в частности, на анализе таких ключевых параметров, как плотность рюкзачного вектора и структурные свойства ассоциированных целочисленных решеток.

Синтез результатов этих двух направлений исследований способствовал формированию более детализированного представления о вычислительной сложности NP-трудных задач, демонстрируя их зависимость от комбинаторных характеристик конкретных экземпляров. Проведенный анализ показал, что практическая трудоемкость решения определяется не только формальной принадлежностью задачи к классу NP-трудных, но и тонкими структурными особенностями ее конкретных экземпляров. Это обусловило актуальность систематической классификации экземпляров задачи на основе их внутренних свойств, включая структуру множества решений, параметры полноты покрытия диапазона допустимых значений и особенности размещения допустимых решений в пространстве поиска.

Таким образом, комплексное исследование задачи о рюкзаке, направленное на строгий анализ ее комбинаторных свойств, создание формальных критериев для классификации экземпляров и оценку вычислительной сложности соответствующих алгоритмов, является актуальным направлением в современной теоретической информатике. Полученные результаты вносят вклад в развитие теории сложности вычислений, дискретной оптимизации и построение новых алгоритмических парадигм, являясь также методологической основой для прикладных исследований в смежных областях.

Объектом исследования являются задачи рюкзачного типа как представители класса NP-полных задач комбинаторной оптимизации.

Предметом исследования служат комбинаторные свойства задач рюкзачного типа и их влияние на вычислительную сложность решения, включая механизмы перехода от легкорешаемых к трудным для решения классам экземпляров при изменении параметров задачи.

Целью работы является исследование взаимосвязи между параметрами и комбинаторной структурой задач рюкзачного типа и их вычислительной сложностью, а также разработка аналитических и алгоритмических методов построения и классификации экземпляров таких задач с заданными свойствами сложности решения.

Для достижения поставленной цели были решены **следующие задачи:**

1. Провести теоретический анализ комбинаторных свойств множества решений задачи о рюкзаке и получить аналитические выражения, связывающие параметры задачи со структурой множества допустимых решений.
2. Исследовать свойства рюкзачных векторов, выделить классы экземпляров с характерными структурными особенностями и построить зависимость вычислительной сложности решения от параметров задачи.
3. Разработать методы построения экземпляров задач с заранее заданными комбинаторными характеристиками, определяющими трудность их решения.
4. Предложить алгоритмы преобразования экземпляров задач рюкзачного типа, позволяющие изменять структуру множества решений и управлять сложностью их решения.
5. Провести алгоритмическую и программную реализацию разработанных методов, подтвердить их эффективность вычислительными экспериментами и продемонстрировать возможности применения в прикладных задачах.

Методы исследования. В исследовании используются аппарат и методы дискретной оптимизации, математической логики, теории информации, теории алгоритмов и сложности вычислений, объектно-ориентированного и логического программирования.

Научная новизна работы заключается в комплексном исследовании комбинаторных и алгоритмических свойств задач рюкзачного типа, развитии

методов их анализа и построении алгоритмов с обоснованной оценкой вычислительной сложности. В ходе исследования получены следующие новые результаты:

1. Найдены и обоснованы новые комбинаторные формулы для анализа множества решений задачи об ограниченном рюкзаке; выведены выражения для среднего числа допустимых решений задач заданной размерности и среднего значения функционала задачи о 0-1 рюкзаке.
2. Введено и исследовано понятие сюръективных линейных форм как специального класса функциональных конструкций; найден алгоритм вычисления всех допустимых решений задач с такими формами и установлены их характеристики, влияющие на сложность вычислений.
3. Разработаны методы построения линейных форм с контролируемым числом решений, позволяющие управлять комбинаторной структурой задачи и конструировать экземпляры с предсказуемым поведением алгоритмов решения.
4. Исследованы линейные формы с разрывами в области допустимых значений и предложен алгоритм их построения; показано, как наличие разрывов влияет на множество допустимых значений и усложняет задачу поиска решения.
5. Разработан алгоритмический метод преобразования легкорешаемых экземпляров задачи о рюкзаке в трудные для решения при сохранении конфигурации множества решений, что открывает возможности применения таких преобразований в задачах защиты информации и анализа сложности дискретных структур.

Обоснованность и достоверность полученных результатов обеспечивается использованием строгого математического аппарата и опорой на фундаментальные исследования в области теории сложности и комбинаторного анализа, представленные в работах отечественных и зарубежных авторов. Предложенные в диссертации теоретические положения подтверждаются строгими доказательствами и согласуются с известными результатами в

смежных областях. Практические алгоритмы и аналитические зависимости проверены вычислительными экспериментами, а основные выводы апробированы на профильных научных конференциях и опубликованы в рецензируемых изданиях.

Теоретическая значимость. В диссертации предложен комплекс научных результатов, формирующих основу для анализа комбинаторных и алгоритмических свойств задач рюкзачного типа. Разработанные методы позволяют:

- проводить формальное описание и классификацию рюкзачных задач по их внутренним параметрам;
- оценивать сложность решения экземпляров задачи и их устойчивость к известным методам решения;
- осуществлять выбор параметров, обеспечивающих заданную сложность решения для конкретных алгоритмов.

Полученные результаты вносят вклад в развитие теории сложности и комбинаторного анализа NP-полных задач.

Практическая значимость работы заключается в разработке алгоритмических средств, позволяющих управлять сложностью экземпляров задач рюкзачного типа и исследовать границы их вычислимости.

В частности:

- предложены методы генерации экземпляров задач с заданными комбинаторными характеристиками, что обеспечивает возможность контроля над числом решений и степенью сложности их нахождения;
- разработан алгоритм поиска всех решений для специального класса рюкзачных задач, обладающий линейной сложностью по числу переменных и количеству решений, что существенно повышает эффективность анализа их структуры;
- предложены преобразования, позволяющие сохранять конфигурацию множества решений при переходе от легкорешаемых к труднорешаемым

экземплярам, что открывает возможности их использования в задачах синтеза и моделирования вычислительно сложных структур.

Полученные результаты могут быть использованы в задачах дискретной оптимизации, анализа сложности алгоритмов, а также при построении схем для систем кодирования и защиты информации.

Положения, выносимые на защиту:

1. Найдены формулы для анализа множества решений задачи об ограниченном рюкзаке, позволяющие вычислять среднее число допустимых решений по всем экземплярам заданной размерности, и среднее значение целевой функции через характеристики подзадач меньшей размерности.
2. Выделен и исследован класс сюръективных линейных форм, для которых множество достижимых значений представляет собой непрерывный диапазон целых чисел; установлены необходимые и достаточные условия сюръективности, а также параметры, определяющие сложность решения определенными алгоритмами. задач с такими формами в качестве левой части.
3. Разработан алгоритм вычисления всех допустимых решений сюръективных линейных форм, обладающий линейной сложностью по числу переменных и количеству решений.
4. Для линейных форм с разрывами в области значений установлены зависимости числа и расположения разрывов от коэффициентов формы, построен алгоритм формирования линейной формы с заданным числом разрывов.
5. Установлены зависимости между комбинаторными параметрами экземпляров задачи о рюкзаке (плотностью, структурой множества решений, диапазоном коэффициентов) и сложностью их решения определенными алгоритмами.
6. Разработаны методы параметрических преобразований экземпляров задачи о рюкзаке, обеспечивающие переход от легкорешаемых к

труднорешаемым случаям при сохранении конфигурации множества решений и показана их применимость в задачах защиты информации.

Апробация результатов. Результаты диссертации докладывались на научных конференциях:

1. XII Международная научно-техническая конференция Безопасные информационные технологии, Москва, 02 ноября 2023 года
2. XIII Международная научно-техническая конференция Безопасные информационные технологии, Москва, 01 ноября 2024 года
3. V Межвузовская конференция аспирантов, соискателей и молодых ученых «Наука, технологии и бизнес», Москва, 18–19 апреля 2023 года

Публикации. Результаты диссертации опубликованы в 9 работах [1-9], из них 5 статей опубликованы в научных журналах, которые включены в перечень рекомендованных ВАК РФ для публикации основных научных результатов диссертаций [1-5], и 3 работы в трудах международных конференций [6-9].

Структура диссертации. Диссертация состоит из введения, трех глав, заключения, списка использованной литературы. Общий объем диссертации составляет 106 страниц, включая 3 таблицы. Список литературы состоит из 77 источников.

Соответствие паспорту специальности. Проведенное исследование соответствует направлениям паспорта специальности 1.2.3 «Теоретическая информатика, кибернетика», в частности: п. 3 «Теория сложности алгоритмов и вычислений» — выполнен систематический анализ вычислительной сложности некоторых классов экземпляров NP-полной задачи о рюкзаке; установлены зависимости сложности решения от комбинаторных параметров задачи и разработаны алгоритмы решения с обоснованными оценками вычислительной сложности; п. 8 «Математическое программирование» — получены аналитические формулы для среднего числа допустимых решений по всем задачам заданной размерности при ограниченных значениях коэффициентов весов и среднего значения функционала задачи о рюкзаке, что позволяет проводить анализ комбинаторных свойств и структуры множества решений

дискретных экстремальных задач; п. 25 «Методы высоконадежной обработки информации и обеспечения помехоустойчивости информационных коммуникаций для целей передачи, хранения и защиты информации» — показано, что предложенные методы позволяют создавать структуры и алгоритмы, основанные на труднорешаемых экземплярах задач о рюкзаке, которые могут использоваться для обеспечения защиты и устойчивости данных при передаче и обработке.

1 Теоретические основы анализа комбинаторных задач рюкзачного типа

Первая глава посвящена формированию теоретических основ исследования задач рюкзачного типа в рамках теоретической информатики и теории сложности вычислений.

Задача о рюкзаке рассматривается как один из базовых примеров NP-полных задач, на основе которых можно анализировать фундаментальные вопросы вычислимости, оценивать границы применимости алгоритмов и выявлять закономерности, определяющие труднорешаемость дискретных оптимизационных задач.

Основное внимание уделено анализу существующих алгоритмических подходов и выявлению пределов их применимости. Рассмотрены методы полного перебора, динамического программирования, приближенные алгоритмы и подходы на основе ветвей и границ.

Сопоставление их свойств показывает, что эффективность алгоритмов определяется не только размерностью задачи, но прежде всего внутренними комбинаторными характеристиками экземпляра — значением коэффициентов, числом допустимых решений и плотностью рюкзачного вектора. Это обстоятельство обосновывает необходимость перехода от описания процедур решения к исследованию структурных свойств самих задач.

В качестве основы для дальнейшего анализа выбран аналитико-комбинаторный подход, позволяющий формально описывать параметры задачи и устанавливать зависимости между структурой входных данных и ее вычислительной сложностью.

Использование производящих функций, методов коэффициентов и анализа комбинаторных характеристик создает теоретическую базу для построения выражений, связывающих параметры задачи с поведением алгоритмов.

Таким образом, в первой главе закладывается теоретический фундамент исследования: формируется понятийный аппарат, уточняются ключевые характеристики задач рюкзачного типа и обосновывается необходимость разработки теории, описывающей их комбинаторные и алгоритмические свойства.

1.1 Задачи рюкзачного типа в теоретической информатике

Теория сложности вычислений составляет фундамент теоретической информатики, изучая требования алгоритмов к ресурсам и принципиальные ограничения вычислимости. В этой области центральное место занимает иерархия классов сложности, среди которых особую роль играют классы P и NP.

Определение 1.1. Класс ***P*** (детерминированное полиномиальное время) состоит из всех задач распознавания, для которых существует детерминированный алгоритм, решающий задачу за время, ограниченное полиномом от длины входа.

Формально: задача Π принадлежит классу P, если существует детерминированная машина Тьюринга M и полином $p(n)$, такие что для любого входа x длины n машина M останавливается за время $O(p(n))$ и возвращает 1 тогда и только тогда, когда $x \in \Pi$.

Определение 1.2. Класс ***NP*** (недетерминированное полиномиальное время) включает задачи распознавания, для которых существует алгоритм верификации решения за полиномиальное от длины входа время [10].

Формально: задача Π принадлежит классу NP, если существует детерминированная машина Тьюринга T и полином $p(n)$, такие что:

- Для любого $x \in \Pi$ существует сертификат y длины $O(p(|x|))$, для которого $T(x, y) = 1$;
- Для любого $x \notin \Pi$ и для любого y выполняется $T(x, y) = 0$;
- Время работы $T(x, y)$ ограничено $O(p(|x|))$.

Классическими примерами задач из класса NP являются [11]:

- **Задача выполнимости булевых формул (SAT)**: определить, существует ли набор значений переменных, обращающий данную булеву формулу в истину;
- **Задача о клике**: определить, содержит ли граф клику заданного размера;
- **Задача о вершинном покрытии**: определить, имеет ли граф вершинное покрытие заданного размера;

Определение 1.3. Задача *П* называется **NP-трудной**, если любая задача из NP сводится к *П* за полиномиальное время.

Определение 1.4. Задача *П* называется **NP-полной**, если:

1. $П \in NP$,
2. *П* является NP-трудной (любая задача из NP сводится к *П* за полиномиальное время).

Фундаментальный вклад в развитие теории сложности внесли Стивен Кук и независимо от него Леонид Левин, доказавшие в 1971-1973 годах существование NP-полных задач [12, 13].

Теорема 1.1 (Кука-Левина). Задача выполнимости булевых формул (SAT) является NP-полной.

Доказательство теоремы состоит из двух частей:

1. Доказательство того, что SAT ∈ NP;
2. Построение полиномиального сведения любой задачи из NP к SAT.

Данный результат предоставил универсальный метод доказательства NP-полноты произвольных задач через сведение к задаче выполнимости булевых формул. Кроме того, эта теорема заложила базис для систематического исследования структуры класса NP, позволив формализовать понятие полноты и установить иерархические отношения между различными комбинаторными задачами. Это открыло новые направления в изучении границ между полиномиально разрешимыми и NP-трудными задачами.

Работа Ричарда Карпа 1972 года [14] расширила этот результат, продемонстрировав NP-полноту 21 комбинаторной задачи, включая:

- Задачу о клике,

- Задачу о вершинном покрытии,
- Задачу о гамильтоновом цикле,
- Задачу о раскраске графа,
- Задачу о точном покрытии.

Для доказательства NP-полноты новой задачи П обычно используется следующая схема [15]:

1. Доказательство принадлежности *П* классу NP:
    - Построение алгоритма верификации сертификата;
    - Доказательство полиномиальности времени верификации.
2. Выбор известной NP-полной задачи *П′*:
    - Как правило, используется SAT или одна из задач из списка Карпа.
3. Построение полиномиального сведения $П' \rightarrow П$:
    - Конструкция преобразования входа;
    - Доказательство эквивалентности: $x \in П' \Leftrightarrow f(x) \in П$;
    - Доказательство полиномиальности преобразования.

Определение 1.5. Полиномиальное сведение задачи $П_1$ к задаче $П_2$ (обозначается $П_1 \propto П_2$ — это функция *f*, вычислимая за полиномиальное время, такая что для любого x выполняется $x \in П_1 \Leftrightarrow f(x) \in П_2$.

Сведение обладает свойствами транзитивности: если $П_1 \propto П_2$ и $П_2 \propto П_3$, то $П_1 \propto П_3$.

Несмотря на теоретическую NP-трудность многих задач, они успешно решаются на практике благодаря нескольким факторам:

1. Приближенные алгоритмы — алгоритмы, находящие решение с гарантированной точностью за полиномиальное время.
2. Эвристические методы — методы, не имеющие теоретических гарантий, но эффективные на практических примерах:
    - Генетические алгоритмы,
    - Методы имитации отжига,
    - Муравьиные алгоритмы.

3. Параметризованная сложность — изучение сложности задач в зависимости от параметров, отличных от размера входа.
4. Изучение частных случаев — выделение классов примеров, разрешимых за полиномиальное время.

Исследование NP-полных задач продолжает оставаться активной областью исследований в теоретической информатике, сочетающей глубокие математические методы с практическими приложениями в оптимизации, искусственном интеллекте и криптографии.

Задача о рюкзаке является одной из фундаментальных задач дискретной оптимизации и классическим примером NP-трудной задачи. Ее изучение имеет двойное значение: с одной стороны, она представляет собой модель для большого числа прикладных задач, связанных с выбором и размещением ресурсов при ограничениях, а с другой — служит удобным инструментом для анализа вычислительной сложности и построения сведений между задачами класса NP [16].

В общем виде задача о 0-1 рюкзаке формулируется следующим образом. Пусть задано множество предметов $I = \{1,2, \dots, n\}$, каждому предмету i соответствует вес $a_i > 0$ и стоимость $c_i > 0$. Требуется выбрать подмножество $S \subseteq I$ так, чтобы суммарный вес выбранных предметов не превышал вместимость рюкзака $b > 0$, а суммарная стоимость была максимальна:

$$\max_{x_i \in \{0,1\}} \sum_{i=1}^{n} c_i x_i \text{ , при условии } \sum_{i=1}^{n} a_i x_i \leq b.$$

Упорядоченный по возрастанию набор весов задачи $A = (a_1, \dots, a_n)$ при этом принято называть рюкзачным вектором, а n – его размерностью.

Это классическая **оптимизационная форма** задачи о рюкзаке. В теории сложности чаще рассматривается **задача о рюкзаке в форме распознавания**, то есть задача определения существования решения, удовлетворяющего заданному ограничению:

существует ли $x_i \in \{0,1\}$, такое, что $\sum_{i=1}^{n} a_i x_i = b$?

Эта форма часто используется при доказательстве NP-полноты, поскольку позволяет четко формулировать вопрос принадлежности к классу NP: для данного вектора x проверка равенства выполняется за полиномиальное время, что удовлетворяет определению класса NP [17].

В зависимости от множества допустимых значений переменных выделяют несколько разновидностей задачи о рюкзаке:

1. **0–1 задача о рюкзаке**, в которой каждый предмет может быть выбран не более одного раза, $x_i \in \{0,1\}$;
2. **задача об ограниченном рюкзаке**, в которой допускается наличие ограниченного количества копий каждого предмета, $x_i \in \{0,1,2,\dots,m_i\}$;
3. **задача о неограниченном рюкзаке**, где число копий не ограничено, $x_i \in \mathbb{Z}_{\geq 0}$.

Задача в форме оптимизации относится к NP-полным, что впервые было доказано через сведение к задаче о рюкзаке в форме распознавания [14]. Таким образом, любая задача из класса NP может быть сведена к ней за полиномиальное время, а следовательно, нахождение эффективного полиномиального алгоритма для задачи о рюкзаке означало бы равенство $P = NP$.

Сложность задачи о рюкзаке зависит не только от числа переменных n, но и от диапазона значений коэффициентов a_i и b. Поскольку эти параметры участвуют в описании входных данных, различают *полиномиальные* и *псевдополиномиальные* алгоритмы. Например, алгоритм динамического программирования, решающий задачу за время $O(nb)$, является полиномиальным относительно числового параметра b, но не по его длине в битовом представлении, что означает, что задача остается NP-трудной [18].

Кроме того, даже при одинаковых формальных параметрах различные конфигурации входных данных могут приводить как к тривиальным, так и к вычислительно трудным случаям. Это обусловлено тем, что значения вектора

весов $A = (a_1, \dots, a_n)$ напрямую влияют на структуру множества допустимых решений и, как следствие, на поведение алгоритмов решения. Таким образом, исследование типов экземпляров задачи о рюкзаке и установление их связи с вычислительной сложностью представляет собой важное направление анализа.

Если коэффициенты a_i образуют строго упорядоченную структуру, обладающую внутренней зависимостью, задача может оказаться решаемой за полиномиальное время. Наиболее известным примером таких упорядоченных структур является сверхрастущая последовательность [19].

Определение 1.6. Вектор $A = (a_1, \dots, a_n)$ называется **сверхрастущим**, если выполняется условие $\forall\, k = 2, \dots, n \;\; a_k > \sum_{i=1}^{k-1} a_i$.

Для задач с таким типом входных данных решение может быть найдено простым жадным алгоритмом, который последовательно выбирает наибольшие возможные элементы, не превышающие целевое значение b. В противоположность этому, равномерно распределенные значения коэффициентов a_i и целевого значения b порождают экземпляры, близкие к худшему случаю. Для таких задач известные точные алгоритмы демонстрируют экспоненциальный рост времени работы с увеличением размерности n [20]. Таким образом, структура входных данных определяет переход от полиномиально решаемых к экспоненциально сложным случаям и играет ключевую роль в классификации экземпляров задачи о рюкзаке.

Важное значение имеет и возможность сведения задачи о рюкзаке к другим NP-полным задачам. Так, она служит базовой при доказательствах NP-полноты для задач упаковки, расписаний, разделения множеств и многих других задач комбинаторной оптимизации. Структурное сходство между этими задачами позволило сформировать целое направление в теоретической информатике, исследующее свойства пространств решений дискретных оптимизационных задач и их комбинаторную топологию [21].

Наряду с теоретической значимостью, задача о рюкзаке широко используется в прикладных исследованиях. Ее модели применяются при

решении задач логистики, распределения ресурсов, построения кодов и в вычислительной криптографии. В последнем случае трудность решения задачи в форме распознавания при случайно выбранных коэффициентах использовалась как основа криптографических схем, что дополнительно стимулировало развитие исследований структуры множества решений [22].

В современных работах теоретическая информатика рассматривает задачу о рюкзаке как типовой объект для анализа границ вычислимости и построения моделей сложности [23]. Поскольку для нее известны как классы легкорешаемых экземпляров, так и классы трудных экземпляров с равномерным распределением коэффициентов, она служит естественной моделью для исследования взаимосвязи структуры входных данных и характеристик вычислительной сложности. Эта особенность делает задачу о рюкзаке центральным примером в изучении NP-полных задач и основой для построения формальных моделей, связывающих комбинаторные свойства с поведением алгоритмов.

Несмотря на простоту постановки, задача о рюкзаке является одной из наиболее исследованных NP-трудных задач, для которой разработан широкий спектр алгоритмических подходов — от точных экспоненциальных методов до приближенных и эвристических схем. Эти подходы различаются не только по вычислительной сложности, но и по принципам использования комбинаторных свойств множества допустимых решений.

Классические методы решения можно условно разделить на четыре группы: переборные алгоритмы, алгоритмы динамического программирования, методы ветвей и границ, а также приближенные и эвристические алгоритмы [24].

Наиболее простым и универсальным способом решения задачи о рюкзаке является *полный перебор* всех возможных комбинаций переменных $x_i \in \{0,1\}$. Общее количество таких комбинаций равно 2^n, поэтому временная сложность такого подхода оценивается как $O(2^n)$. Этот метод гарантирует нахождение оптимального решения, однако его применение ограничено малыми размерностями задачи.

Формально, алгоритм полного перебора проверяет все подмножества множества предметов $I = \{1,2, \dots, n\}$, вычисляя для каждого подмножества $S \subseteq I$ значения

$$A(S) = \sum_{i \in S} a_i \, , C(S) = \sum_{i \in S} c_i \, ,$$

и выбирает то S^*, для которого $A(S^*) \leq b$ и $C(S^*)$ максимально. Несмотря на экспоненциальную сложность, метод полного перебора служит основой для построения более эффективных схем, в частности, алгоритмов ветвей и границ.

Вариацией этого подхода является метод *инкрементального перебора*, при котором пространство решений упорядочивается и сокращается за счет отсечения невозможных комбинаций на ранних этапах. Такие методы позволяют уменьшить среднее время решения, но не изменяют асимптотику в худшем случае.

Для задачи о рюкзаке возможна постановка в рекурсивной форме, что позволяет применять метод *динамического программирования* [25]. Основное соотношение Беллмана имеет вид:

$$f(i,b) = \begin{cases} 0, & i = 0, \\ f(i-1,b), & a_i > b, \\ \max\left(f(i-1,b), f(i-1,b-a_i) + c_i\right), & a_i \leq b, \end{cases}$$

где $f(i,b)$ обозначает максимальную стоимость, достижимую с использованием первых i предметов при ограничении вместимости b. Рекурсивная структура задачи позволяет построить решение в виде таблицы размером $n \times b$, что дает временную сложность $O(nb)$ и пространственную $O(nb)$.

Такой алгоритм является *псевдополиномиальным*, поскольку его сложность зависит линейно от значения параметра b, но не от его длины в битах. Это означает, что если b велико, то время работы алгоритма растет экспоненциально по длине входных данных. Тем не менее метод динамического программирования является одним из наиболее эффективных точных алгоритмов при умеренных значениях b и широко используется на практике [26].

Существует также модификация данного подхода для ограниченной задачи о рюкзаке, где для каждого предмета задано ограничение m_i на количество возможных копий. В этом случае используется преобразование каждой переменной x_i в двоичное представление, что позволяет свести задачу к эквивалентной 0–1 форме при сохранении порядка сложности [16].

Метод ветвей и границ [27] представляет собой усовершенствование переборного подхода за счет систематического исключения заведомо невыгодных ветвей пространства решений. Основная идея заключается в том, что каждая частичная комбинация переменных рассматривается как узел дерева поиска, которому сопоставляется верхняя оценка (граница) возможного значения целевой функции. Если верхняя оценка для текущей ветви меньше, чем уже найденное лучшее решение, ветвь отсекается.

На практике границы вычисляются с использованием *жадной оценки*, то есть предполагается, что оставшиеся предметы могут быть добавлены дробным образом (аналогично релаксации задачи). Для раннего отсечения узлов используется приближенное решение, полученное жадным алгоритмом, что существенно сокращает размер дерева поиска.

Временная сложность метода ветвей и границ варьируется от $O(n)$ в благоприятных случаях до $O(2^n)$ в худшем, однако на реальных данных он часто показывает существенно лучшие результаты за счет высокой эффективности отсечения.

Поскольку задача о рюкзаке является NP-трудной, большое внимание уделяется построению приближенных и эвристических алгоритмов, обеспечивающих допустимые, но не обязательно оптимальные решения. Среди них наиболее известен *жадный алгоритм* [28], основанный на выборе предметов в порядке убывания отношения c_i/a_i. Этот алгоритм дает точное решение для дробной (линейно релаксированной) задачи и приближенное — для дискретной 0–1 формы.

Для задачи о рюкзаке существует также *приближенная схема полиномиального времени (FPTAS)*, которая для любого $\varepsilon > 0$ находит решение с гарантией

$$C^* \leq C_\varepsilon \leq (1+\varepsilon)C^*,$$

где C^*— оптимальное значение, а время работы оценивается как $O(n^3/\varepsilon)$ [29]. Эти результаты особенно важны в контексте теории сложности, поскольку показывают, что несмотря на NP-трудность задачи, существуют приближенные алгоритмы с контролируемой точностью.

Современные исследования включают также метаэвристические методы — генетические алгоритмы, методы имитации отжига, поиск с запретами и методы роя частиц, применяемые для больших размерностей n. Эти методы не дают гарантированной точности, но обладают высокой практической эффективностью и часто используются в прикладных задачах, связанных с оптимизацией ресурсов [30].

Таким образом, разнообразие алгоритмических подходов к решению задачи о рюкзаке отражает ее фундаментальное значение для теоретической информатики. Методы динамического программирования и ветвей и границ обеспечивают точные решения для задач умеренной размерности, а приближенные и эвристические алгоритмы — эффективные решения для крупных экземпляров. Эти алгоритмические подходы, основанные на анализе структуры множества допустимых решений, составляют основу современных методов исследования вычислительной сложности и моделирования трудных комбинаторных задач.

1.2 Аналитические методы исследования комбинаторных структур

Несмотря на значительное число известных алгоритмов решения задачи о рюкзаке — от точных переборных и динамических до приближенных и декомпозиционных, — их эффективность в реальных вычислительных сценариях во многом определяется структурными свойствами конкретных экземпляров задачи. Даже в пределах одного класса постановок наблюдаются

существенные различия в вычислительной сложности, обусловленные взаимным расположением коэффициентов, их диапазоном и плотностью вектора весов [31].

Определение 1.7. Плотностью рюкзачного вектора $A = (a_1, a_2, \dots, a_n)$, состоящего из положительных целых чисел, называется величина $d(A) = \frac{n}{\log_2(\max_i a_i)}$, характеризующая соотношение между размерностью вектора и битовой длиной его максимального элемента.

Плотность рюкзака тесно связана со структурой множества допустимых решений и во многом определяет поведение алгоритмов при поиске оптимума. Известно, что при малой плотности задача, как правило, решается значительно быстрее, тогда как при $d(A) \approx 1$ достигается область наибольшей вычислительной трудности [32]. В этом контексте особый интерес представляет анализ асимптотических и комбинаторных характеристик пространства решений, позволяющих формально описывать зависимость времени работы алгоритмов от параметров входных данных.

Отдельное направление исследований связано с изучением вычислительной сложности в частных случаях и разработкой оценок трудоемкости конкретных алгоритмов при фиксированных структурах входных данных. В рамках этого подхода ключевое место занимает метод ветвей и границ, который позволяет формально связать структуру задачи с количеством рассматриваемых подзадач и оценить ожидаемое время работы алгоритма. Значительный вклад в развитие данного направления внесли отечественные исследователи Р. М. Колпаков и М. А. Посыпкин, систематически исследовавшие поведение метода ветвей и границ применительно к задаче о рюкзаке и ее вариациям. В их работах [33, 34] предложены аналитические оценки числа шагов метода через параметры исходных данных, что позволило выделить классы экземпляров, для которых метод демонстрирует квазиполиномиальную зависимость времени работы от размерности задачи.

В последующих исследованиях [35, 36] рассматривались модификации метода с улучшенными стратегиями ветвления и отсечения, позволяющие

учитывать не только численные параметры задачи, но и комбинаторную структуру множества решений. Эти подходы продемонстрировали, что даже в пределах одной NP-трудной задачи возможно выделение подклассов экземпляров, обладающих существенно различной вычислительной сложностью.

Дальнейшее развитие этого направления связано с переходом от анализа алгоритмов к анализу самих задач, то есть к исследованию закономерностей, определяющих количество и структуру множества допустимых решений в зависимости от параметров входных данных. Такой подход позволяет получить оценки не только для конкретных алгоритмов, но и для класса методов в целом, а также выявить области параметров, в которых задача достигает максимальной комбинаторной сложности [37].

Особое значение имеет изучение среднего числа решений задачи о рюкзаке при различных распределениях коэффициентов, что позволяет количественно описать плотность пространства допустимых решений. Использование производящих функций для описания этого множества позволяет выразить комбинаторные характеристики через рекуррентные зависимости от размерности задачи и диапазона значений коэффициентов. Подобные методы применимы не только для оценки вероятности существования решения, но и для анализа поведения эвристических и приближенных алгоритмов.

Таким образом, современный подход к исследованию задачи о рюкзаке выходит за рамки анализа отдельных алгоритмов. В центре внимания оказывается внутренняя структура задачи — множество решений, свойства рюкзачных векторов, взаимосвязь между плотностью и вычислительной сложностью. Этот сдвиг акцентов от процедурного к структурному анализу открывает возможности для построения новых теоретических оценок и разработки алгоритмов, адаптированных под конкретные классы экземпляров.

Метод производящих функций является одним из наиболее мощных аналитических инструментов в комбинаторном анализе. Его суть заключается в том, что комбинаторная структура — множество объектов, упорядоченных по

некоторому параметру, — представляется в виде степенного ряда, коэффициенты которого отражают численные характеристики этих объектов. Такой подход позволяет использовать методы математического анализа для получения точных и асимптотических оценок, нахождения рекуррентных соотношений и решения широкого класса дискретных задач [38].

Пусть дана последовательность чисел

$$\{a_n\}_{n=0}^{\infty}, a_n \in \mathbb{R}.$$

Обычной производящей функцией (ordinary generating function) этой последовательности называется формальный степенной ряд

$$A(u) = \sum_{n=0}^{\infty} a_n u^n$$

Если последовательность a_n является конечной, то данный ряд представляет собой многочлен, а если бесконечной — формальный ряд, не требующий сходимости в аналитическом смысле.

Производящие функции позволяют записывать в компактной форме сложные комбинаторные зависимости и операции над последовательностями. Так, операции над рядами — сложение, умножение, дифференцирование, замена переменных — естественным образом соответствуют операциям над комбинаторными объектами, например, объединению или конкатенации множеств.

Одной из центральных идей метода производящих функций является метод коэффициентов, подробно разработанный в работах В. А. Егорычева [39]. Метод коэффициентов представляет собой вариант формализма производящих функций, основанный на определении линейного функционала на множестве формальных степенных рядов, содержащих лишь конечное число членов с отрицательными степенями.

Обозначим через $C(u) = \{A(u)\}$ класс таких рядов:

$$A(u) = u^k(a_0 + a_1 u + a_2 u^2 + \cdots), a_0 \neq 0,\ k = 0, \pm 1, \pm 2, \ldots$$

Для ряда $A(u)$ определим оператор выбора коэффициента:

$$\underset{u}{coef}\{A(u)\} = a_{-1}.$$

Иными словами, оператор $\underset{u}{coef}$ выделяет коэффициент при u^{-1} в формальном ряде $A(u)$. Таким образом, на множестве $C(u)$ определен линейный функционал, который ставит в соответствие каждому ряду его коэффициент при u^{-1}.

Основные свойства этого функционала [40] формулируются следующим образом:

1. **Линейность**

$$\underset{u}{coef}\{\lambda A(u) + \mu B(u)\} = \lambda \underset{u}{coef}\{A(u)\} + \mu \underset{u}{coef}\{B(u)\}.$$

2. **Замена переменной**

Если $A(u)$ — степенной ряд, не содержащий отрицательных степеней u, то

$$\sum_{k=0}^{\infty} z^k \underset{u}{coef}\{A(u)u^{-k-1}\} = A(z). \quad (1)$$

Это свойство обеспечивает переход от формального функционала к обычной производящей функции и позволяет использовать интегральные методы анализа.

3. **Интегральная форма**

Если ряд $A(u)$ сходится в окрестности нуля, то

$$\underset{u}{coef}\{A(u)\} = \frac{1}{2\pi i}\oint_{|u|=\rho} A(u)\,du = \operatorname{res}_{u=0}\{A(u)\}.$$

Таким образом, оператор $\underset{u}{coef}$ эквивалентен взятию вычета функции $A(u)$ при $u = 0$, что открывает возможность использовать методы комплексного анализа в комбинаторных вычислениях.

Метод коэффициентов позволяет не только формализовать операции над производящими функциями, но и получить целый ряд полезных тождеств. Например:

$$\underset{u}{coef}\{(1+u)^n u^{-k-1}\} = \binom{n}{k}, \quad (2)$$

$$\underset{u}{coef}\left\{e^{\lambda u}u^{-k-1}\right\} = \frac{\lambda^k}{k!},$$

$$\underset{u}{coef}\{\ln(1-u)\,u^{-k-1}\} = -\frac{1}{k},$$

$$\underset{u}{coef}\{(1-u)^{-n}u^{-k-1}\} = \binom{n+k-1}{k}. \quad (3)$$

Эти формулы составляют основу для анализа комбинаторных сумм, подсчета числа решений диофантовых уравнений и построения аппроксимаций.

Метод коэффициентов особенно эффективен при вычислении комбинаторных сумм и при анализе ограниченных областей суммирования. Его использование позволяет формализовать переход от комбинаторной задачи к аналитическому выражению, где под знаком оператора $\underset{u}{coef}$ оказывается функция, отражающая структуру исходной задачи.

Как отмечается в [41], применение метода коэффициентов обеспечивает единый аналитический язык для описания широкого круга задач — от анализа рекуррентных соотношений до вычисления сложных вероятностных характеристик. В частности, метод используется для построения рациональных производящих функций, выражающих структуру множеств допустимых решений в задачах дискретной оптимизации.

Таким образом, метод коэффициентов служит естественным связующим звеном между комбинаторным и аналитическим подходами. Он позволяет не только вычислять конкретные комбинаторные величины, но и выявлять общие закономерности в структуре дискретных задач, что делает его одним из базовых инструментов современной теоретической информатики.

1.3 Постановка задачи исследования и обоснование выбранного подхода

Задача о рюкзаке, как одна из базовых NP-трудных задач, является универсальной моделью для широкого класса дискретных задач оптимизации и комбинаторных конфигураций. Несмотря на многолетнюю историю ее изучения, остаются открытыми вопросы, касающиеся структуры пространства

допустимых решений, характера распределения функционала и взаимосвязи между параметрами задачи и ее вычислительной сложностью. Эти вопросы выходят за рамки сугубо алгоритмического анализа и требуют привлечения аналитических и комбинаторных методов исследования.

Современные методы анализа комбинаторных задач все чаще опираются на структурные характеристики множеств решений, а не только на поведенческие оценки конкретных алгоритмов. В задаче о рюкзаке такая структура определяется множеством достижимых значений сумм $\sum a_i x_i$ при $x \in \{0,1\}^n$.

Понимание закономерностей этого структуры этого множества имеет фундаментальное значение для:

- оценки средней и предельной вычислительной сложности различных алгоритмических подходов;
- анализа устойчивости решений к возмущениям исходных данных;
- построения параметризованных классов задач с контролируемыми свойствами (например, плотностью, числом допустимых решений, симметрией решений);
- обоснования применимости задач рюкзачного типа как базовых моделей в прикладных областях — от теории кодирования до криптографических систем.

Существенную трудность представляет то, что для большинства NP-трудных задач, включая задачу о рюкзаке, распределение значений функционала не имеет простой аналитической формы. Для типичных случайных экземпляров оно приближается к нормальному, однако при структурных ограничениях наблюдаются закономерные отклонения от симметрии и плотности, которые напрямую влияют на поведение алгоритмов.

Чем ниже плотность, тем легче некоторые алгоритмы редукции базиса решеток (например LLL-алгоритм) решают задачу, поскольку существует больше «пространства» для линейных зависимостей между векторами базиса решетки. Когда же плотность приближается к некоторым пороговым значениям порядка

0,94, экземпляры становятся гораздо более устойчивыми к таким методам, и время решения резко возрастает. В литературе показано, что существует критический порог плотности, ниже которого LLL-редукция становится эффективной, и выше которого ее эффективность убывает экспоненциально [42, 43].

Механизм решения задачи о рюкзаке при низкой плотности основан на ее сведении к задаче поиска короткого вектора в целочисленной решетке. Фундаментальную роль здесь играет алгоритм Ленстры-Ленстры-Ловаса [44].

Определение 1.8. Решеткой в $\mathbb{R}^n$ называется дискретная подгруппа вида:

$$\mathcal{L} = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \right\}$$

где $\{b_1, \dots, b_n\}$ — базис решетки.

Определение 1.9. Алгоритм LLL (Ленстры-Ленстры-Ловаса) — это полиномиальный алгоритм редукции базиса евклидовой решетки. Он преобразует произвольный базис решетки

$$B = (b_1, b_2, \dots, b_n)$$

в редуцированный базис

$$B' = (b_1', b_2', \dots, b_n'),$$

в котором векторы являются более короткими и почти ортогональными, сохраняя при этом ту же решетку.

Алгоритм основан на модифицированной ортогонализации Грама–Шмидта и последовательном приведении векторов к форме, удовлетворяющей условиям редукции:

$$| \mu_{i,j} | \leq \frac{1}{2} \text{для всех } i > j,$$

и

$$\| b_i^* \|^2 \leq (\delta - \mu_{i,i-1}^2) \ \| b_{i-1}^* \|^2, \delta \in (\frac{1}{4}, 1),$$

где b_i^*— ортогонализованные векторы по Граму–Шмидту, а $\mu_{i,j}$— коэффициенты ортогонализации.

Результатом работы алгоритма является δ-редуцированный базис решетки, который может использоваться для приближенного решения задач поиска короткого вектора (SVP) и ближайшего вектора (CVP).

Для решения задачи о рюкзаке с вектором весов $A = (a_1, \dots, a_n)$ и целевым значением b строится решетка $\mathcal{L} \subset \mathbb{Z}^{n+1}$, порожденная строками матрицы:

$$B = \begin{pmatrix} 1 & 0 & \cdots & 0 & Na_1 \\ 0 & 1 & \cdots & 0 & Na_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & Na_n \\ 0 & 0 & \cdots & 0 & -Nb \end{pmatrix}$$

где N — достаточно большое целое число.

Теорема 1.2. (Лагариас, Одлыжко, 1985). Если задача о рюкзаке имеет решение и плотность $d(A) < 0{,}64$, то с высокой вероятностью алгоритм LLL, примененный к решетке $\mathcal{L}$, найдет вектор, соответствующий решению исходной задачи [45].

Впоследствии результаты Лагариаса и Одлыжко были существенно уточнены и расширены. В работе Костера и др. [46] предложен улучшенный алгоритм решения задач о сумме подмножеств на основе методов редукции решеток. Основное усовершенствование состояло в итеративном применении алгоритма LLL в сочетании с модифицированными процедурами отбора векторов и масштабирования. Этот подход позволил повысить эффективность восстановления коротких векторов и существенно расширить диапазон плотностей, при которых метод остается применимым. Авторы показали, что усовершенствованный алгоритм способен успешно решать экземпляры задачи с плотностью $d(A) < 0{,}94$, что значительно превосходит прежнюю границу $d(A) < 0{,}64$, установленную в теореме Лагариаса и Одлыжко.

Дальнейшее развитие методов решения задач на решетках связано с переходом от редукционных алгоритмов к поисковым схемам на основе просеивания (sieving). В работе [47] предложен алгоритм для задачи нахождения кратчайшего вектора (SVP), представляющий собой вероятностный метод, основанный на последовательном построении и фильтрации множеств векторов

с использованием стохастических эвристик. Этот алгоритм продемонстрировал, что методы просеивания могут превосходить классические процедуры типа LLL при решении задач высокой размерности, особенно для решеток с равномерным распределением длин векторов.

Результаты [48] и [49] в совокупности определили современное понимание границ применимости методов редукции решеток в анализе задач о рюкзаке. Они показали, что вычислительная сложность решения конкретных экземпляров таких задач зависит не только от их формальной NP-трудности, но и от величины коэффициентов, плотности и распределения возможных решений, что делает исследование этих параметров принципиально важным для оценки сложности задач данного типа.

В связи с этим возникает основная научная задача — разработать аналитический аппарат для описания и оценки комбинаторных свойств задачи о рюкзаке, в частности, числа и структуры допустимых решений, а также параметров, определяющих переход от легкоразрешимых к трудным экземплярам.

В качестве основного инструмента исследования выбран аппарат производящих функций и метода коэффициентов, развитый в работах [39, 40]. Эти методы позволяют перейти от дискретного описания множества решений к его аналитическому представлению в виде функционалов и интегральных выражений, пригодных для асимптотического анализа.

Ключевое преимущество такого подхода состоит в возможности формализованного описания не только единичных решений, но и их совокупных характеристик — среднего числа, распределения, плотности и корреляционных зависимостей между элементами.

В отличие от чисто алгоритмических или вероятностных моделей, аналитический подход позволяет получать универсальные выражения, применимые к широкому спектру классов задач — от классической задачи о рюкзаке до ее вариаций с ограничениями и специальными структурами коэффициентов.

Для систематизации исследования были выделены следующие направления анализа:

1. Формулировка аналитических представлений множества решений через производящие функции и установление взаимосвязи между их коэффициентами и комбинаторными характеристиками задачи.
2. Выведение формул для среднего числа решений и его дисперсии в зависимости от параметров задачи (размерности, диапазона весов, плотности).
3. Исследование асимптотических свойств полученных выражений и установление критериев перехода от низкоплотных к высокоплотным случаям.
4. Построение алгоритмических аналогов аналитических оценок для практического вычисления характеристик на ограниченных выборках.

Таким образом, выбранный подход сочетает аналитическую строгость комбинаторных методов с вычислительной реализуемостью.

Предлагаемый в работе подход позволяет рассматривать задачу о рюкзаке не как изолированную NP-трудную задачу, а как элемент единой системы комбинаторных структур, допускающих аналитическое описание.
В отличие от классических методов, где анализ сосредоточен на поиске оптимального решения, акцент делается на структуре пространства допустимых решений, что открывает возможности для:

- вывода точных и приближенных формул для количества решений в зависимости от параметров задачи;
- построения комбинаторных оценок сложности и критериев «трудности» экземпляров;
- разработки универсальных параметрических моделей, пригодных для применения в других областях дискретной математики и теоретической информатики.

В частности, аналитические соотношения, полученные в дальнейшем, позволяют формализовать переход от отдельных задач оптимизации к их

агрегированным характеристикам. Такой подход является необходимым шагом для построения обобщенных моделей, описывающих не только вычислительную, но и структурную сложность комбинаторных объектов.

Развитие методов анализа комбинаторных структур, объединяющих дискретные и аналитические аспекты, отражает современную тенденцию интеграции теоретической информатики, алгебраической комбинаторики и математического анализа.

Метод производящих функций и связанных с ним операторных преобразований является признанным инструментом исследования таких структур и позволяет не только уточнять известные результаты, но и формулировать новые типы параметрических зависимостей.

Особое значение предложенный подход имеет для задач, где требуются контролируемые комбинаторные свойства, например, при синтезе тестовых выборок, моделировании вероятностных распределений или генерации параметров алгоритмов.

В том числе такие результаты находят применение в криптографических конструкциях, где свойства комбинаторных задач используются для построения стойких преобразований, хотя в рамках настоящей работы акцент делается именно на теоретико-комбинаторной стороне анализа.

Выводы к главе 1

В первой главе сформулированы теоретические основания исследования задач о рюкзаке в контексте теоретической информатики и теории сложности вычислений. Показано, что задача о рюкзаке представляет собой один из канонических примеров NP-полных задач, служащий удобной моделью для изучения границ вычислимости и структурных свойств дискретных оптимизационных задач.

Анализ известных алгоритмических подходов — переборных, динамических, приближенных и метода ветвей и границ — продемонстрировал ограниченность классических методов и необходимость перехода от чисто

алгоритмического анализа к исследованию внутренних комбинаторных свойств экземпляров задачи. Именно структура множества допустимых решений, распределение коэффициентов и плотность рюкзачного вектора во многом определяют вычислительную сложность и поведение алгоритмов.

На основе анализа литературы и выявленных закономерностей была поставлена задача разработки теоретических и аналитических методов исследования комбинаторных структур задач о рюкзаке. В качестве методологической основы выбран аналитико-комбинаторный подход, включающий использование производящих функций, метода коэффициентов и анализа комбинаторных характеристик задачи (таких как плотность, число решений, вид линейных форм). Эти средства позволяют перейти от описания алгоритмов к формальному анализу внутренней структуры задач и построению выражений, связывающих параметры входных данных с вычислительной сложностью.

Таким образом, в первой главе определено направление дальнейшего исследования: переход от изучения частных алгоритмов решения задачи о рюкзаке к разработке общей теории, описывающей комбинаторные закономерности ее структуры и обеспечивающей формальный базис для оценки сложности и построения алгоритмов нового типа.

2 Комбинаторный анализ и построение линейных форм с управляемыми характеристиками

В данной главе исследуются комбинаторные свойства множества решений задачи о рюкзаке и их влияние на вычислительную сложность. Основное внимание уделяется анализу взаимосвязи параметров задачи и структуры множества допустимых решений. С использованием метода производящих функций получены аналитические выражения, позволяющие вычислять среднее число допустимых решений и среднее значение целевой функции через параметры задачи — размерность, диапазон коэффициентов и правую часть неравенства.

Особое внимание уделяется линейным формам специального вида, обозначенными в работе как сюръективные формы. Эти формы представляют собой примеры структур, для которых множество допустимых решений покрывает всю область возможных значений правой части уравнения. Благодаря такому свойству, для задач с сюръективными формами существует эффективная процедура нахождения всех решений, линейно зависящая от числа переменных и числа решений. Однако при модульных преобразованиях, применяемых к коэффициентам формы, теряется свойство сюръективности, и соответствующие задачи становятся трудными для решения известными алгоритмами, при этом сохраняя конфигурацию множества решений. Такое поведение делает сюръективные формы важным объектом для анализа границ применимости алгоритмов и изучения влияния структурных свойств задачи на ее алгоритмическую сложность.

Кроме того, в данной главе рассматриваются линейные формы с разрывами в области значений, позволяющие формализовать процесс утраты сюръективности и исследовать механизмы перехода от простых к трудным для решения экземплярам.

2.1 Комбинаторные свойства задачи о рюкзаке

Для доказательства основных результатов в данном разделе использован метод производящих функций. Базовой техникой для выражения ограничений на множество допустимых решений послужил метод коэффициентов [39]. Данный метод определяет линейный функционал на множестве формальных степенных рядов с конечным числом членов отрицательной степени, который ставит в соответствие каждому степенному ряду коэффициент при его члене минус первой степени. Для степенных рядов, сходящихся в окрестности нуля, этот коэффициент совпадает с вычетом в точке 0. В ряде случаев этот метод существенно удобнее классического варианта с применением вычетов.

Для большей общности рассмотрим задачу об ограниченном рюкзаке, которая обобщает классическую постановку задачи о 0–1 рюкзаке и позволяет каждой переменной принимать несколько дискретных значений. В форме задачи целочисленного программирования она записывается как [16]:

$$\sum_{j=1}^{n} c_j x_j \to max \tag{4}$$

$$\sum_{i=1}^{n} a_i x_i \leq b \tag{5}$$

где $x = (x_1, \dots, x_n)$ – n-мерный вектор с целочисленными компонентами $x_i \in \{0, 1, \dots, m\}$,

$c_1, \dots, c_n;\ a_1, \dots, a_n;\ b$ – неотрицательные целые числа.

Выразим производящие функций в виде полиномов для множества допустимых решений и значений функционала задачи на этом множестве. Множество допустимых решений задачи V_b – это множество n-мерных векторов x с $x_i \in \{0, 1, \dots, m\}, i = 1, \dots, n$, удовлетворяющих неравенству (5). Объемом множества допустимых решений V_b назовем число $|V_b|$ допустимых решений неравенства (5).

Для анализа распределения точек на множестве допустимых решений задачи (5) используется полином

$$P_b(z_1, \dots, z_n) = \sum_{x \in V_b} z_1^{a_1 x_1} z_2^{a_2 x_2} \dots z_n^{a_n x_n} \qquad (6)$$

Для исследования свойств значений функционала задачи (5) в допустимых точках множества решений будет рассматриваться полином

$$F_b(z_1, \dots, z_n) = \sum_{x \in V_b} z_1^{c_1 x_1} z_2^{c_2 x_2} \dots z_n^{c_n x_n} \qquad (7)$$

Примеры использования этих полиномов для получения оценок в различных типах задачи о рюкзаке приведены в работах [50] и [51].

Лемма 2.1. Для задачи об ограниченном рюкзаке (4), (5) справедлива формула

$$\sum_{b=0}^{\infty} P_b(z_1, \dots, z_n)\, u^b = \\ = \frac{(1 + (z_1 u)^{a_1} + \dots + (z_1 u)^{m a_1}) \dots (1 + (z_n u)^{a_n} + \dots + (z_n u)^{m a_n})}{1 - u}. \qquad (8)$$

Доказательство. Преобразуем сумму (8), используя метод коэффициентов. Внутреннее суммирование проводится по всему множеству векторов $(x_1, x_2, \dots, x_n)$ с координатами из {0, 1, …, m}. Использование метода коэффициентов позволяет отбирать из этого множества только те векторы, которые удовлетворяют ограничениям (5).

$$P_b(z_1, \dots, z_n) = \sum_{t=0}^{b} \sum_{\{x_1, \dots, x_n\}} z_1^{a_1 x_1} z_2^{a_2 x_2} \dots z_n^{a_n x_n} \underset{u}{coef} \left\{ \frac{u^{\sum_{i=1}^{n} a_i x_i}}{u^{t+1}} \right\}$$

Здесь и далее $\underset{u}{coef}\{A(u)\} = \frac{1}{2\pi i} \oint_{|u|=\rho} A(u) du = a_{-1}$

где a_{-1} – коэффициент при минус первой степени многочлена $A(u)$,

ρ – параметр $0 < \rho < 1$.

Подробное описание данного функционала и его свойств приведено в [39].

Занося значения, зависящие от x, под знак коэффициента и объединяя множители с одинаковыми показателями, получим

$$P_b(z_1, \dots, z_n) = \underset{u}{coef} \left\{ \sum_{t=0}^{b} \frac{1}{u^{t+1}} \sum_{x_1=0}^{m} (z_1 u)^{a_1 x_1} \dots \sum_{x_n=0}^{m} (z_n u)^{a_n x_n} \right\} =$$

$$= \underset{u}{coef}\left\{\frac{\frac{1}{u}-\frac{1}{u^{b+2}}}{1-\frac{1}{u}}\prod_{k=1}^{n}(1+(z_k u)^{a_k}+\cdots+(z_k u)^{ma_k})\right\}.$$

Раскладывая полученное выражение по числителю дроби, имеем

$$P_b(z_1,\dots,z_n) = \underset{u}{coef}\left\{\frac{-1}{(1-u)}\prod_{k=1}^{n}(1+(z_k u)^{a_k}+\cdots+(z_k u)^{ma_k})\right\}+$$

$$+\underset{u}{coef}\left\{\frac{1}{u^{b+1}(1-u)}\prod_{k=1}^{n}(1+(z_k u)^{a_k}+\cdots+(z_k u)^{ma_k})\right\}.$$

Теперь заметим, что ввиду теоремы о вычетах [39], первое слагаемое равно нулю и получим

$$P_b(z_1,\dots,z_n) = \underset{u}{coef}\left\{\frac{1}{u^{b+1}(1-u)}\prod_{k=1}^{n}(1+(z_k u)^{a_k}+\cdots+(z_k u)^{ma_k})\right\}. \qquad (9)$$

Подставляя выражение (9) в левую часть формулы (8), получим

$$\sum_{b=0}^{\infty} P_b(z_1,\dots,z_n)\,u^b =$$

$$= \sum_{b=0}^{\infty} u^b \underset{u}{coef}\left\{\frac{1}{u^{b+1}(1-u)}\prod_{k=1}^{n}(1+(z_k u)^{a_k}+\cdots+(z_k u)^{ma_k})\right\}.$$

Теперь, воспользовавшись правилом замены переменной (1) для u, получим искомое соотношение.

Следствие 2.1. Для объема области допустимых решений задачи (4), (5) с $m \in \mathbb{N}$ имеет место равенство

$$|V_b| = \underset{u}{coef}\left\{\frac{(1+u^{a_1}+\cdots+u^{ma_1})\dots(1+u^{a_n}+\cdots+u^{ma_n})}{(1-u)u^{b+1}}\right\}. \qquad (10)$$

Доказательство. Для нахождения числа допустимых решений задачи необходимо подставить $z = 1$ в ряд (6) и полученное выражение подставить в (9).

Лемма 2.2. Имеет место равенство:

$$F_b(z_1,\dots,z_n) =$$

$$= \underset{u}{coef}\left\{\frac{\left(1+z_1^{c_1}u^{a_1}+\cdots+z_1^{mc_1}u^{ma_1}\right)\ldots\left(1+z_n^{c_n}u^{a_n}+\cdots+z_n^{mc_n}u^{ma_n}\right)}{(1-u)u^{b+1}}\right\}. \quad (11)$$

Доказательство. Преобразуем сумму (7), используя метод коэффициентов. Аналогично предыдущей лемме, введение функционала $\underset{u}{coef}$ позволит получить ограничение области допустимых решений задачи.

$$F_b(z_1,\ldots,z_n) = \sum_{t=0}^{b}\sum_{\{x_1,\ldots,x_n\}} z_1^{c_1x_1}\, z_2^{c_2x_2}\ldots z_n^{c_nx_n}\ \underset{u}{coef}\left\{\frac{u^{\sum_{i=1}^{n}a_ix_i}}{u^{t+1}}\right\} =$$

$$= \underset{u}{coef}\left\{\sum_{t=0}^{b}\frac{1}{u^{t+1}}\sum_{x_1=0}^{m}\left(z_1^{c_1}u^{a_1}\right)^{x_1}\ldots\sum_{x_n=0}^{m}\left(z_n^{c_n}u^{a_n}\right)^{x_n}\right\} =$$

$$= \underset{u}{coef}\left\{\frac{1}{u^{b+1}(1-u)}\prod_{k=1}^{n}\left(1+z_k^{c_k}u^{a_k}+\cdots+z_k^{mc_k}u^{ma_k}\right)\right\}. \quad (12)$$

Результаты этих утверждений будут применены в дальнейших примерах и построениях.

Для эффективного решения задачи о рюкзаке при помощи алгоритмов декомпозиции и перебора необходимо иметь способы оценки значений функционала решений задачи. В этом контексте может быть полезна формула, которая выражает среднее значение функционала на множестве допустимых решений.

Рассмотрим производящую функцию (7), которая характеризует распределение значений функционала $f(x_1,\ldots,x_n) = \sum_{j=1}^{n} \mathrm{c}_j x_j$ рассматриваемой задачи (4), (5). Для некоторого целого неотрицательного k обозначим A_k - число допустимых решений задачи, в которых значение целевой функции $f(x_1,\ldots,x_n) = \sum_{j=1}^{n} \mathrm{c}_j x_j$ равно k.

Также введем следующее обозначение:

$$\Phi_b(z) = F_b(z,\ldots,z) = \sum_{x\in V_b} z^{c_1x_1}\, z^{c_2x_2}\ldots z^{c_nx_n} = \sum_{k=0}^{\infty} A_k\, z^k. \quad (13)$$

Из введенных ранее определений, обозначений и формулы (13) следует соотношение

$$|V_b| = \Phi_b(1) = F_b(1,\ldots,1) = \sum_{x\in V_b} 1^{c_1x_1}\,1^{c_2x_2}\ldots 1^{c_nx_n} = \sum_{k=0}^{\infty} A_k.$$

Обозначим $f(x_1,\ldots,x_n) = \sum_{j=1}^{n} \mathrm{c}_j x_j$. В частности, заметим, что

$$\max_{x\in V_b} f(x_1,\ldots,x_n) = \max_{x\in V_b} \sum_{j=1}^{n} \mathrm{c}_j x_j = \max_{k:\, A_k\geq 1} k.$$

Далее из леммы 2 получим формулу

$$\Phi_b(z) =$$
$$= \underset{u}{coef}\left\{\frac{(1+z^{c_1}u^{a_1}+\cdots+z^{mc_1}u^{ma_1})\ldots(1+z^{c_n}u^{a_n}+\cdots+z^{mc_n}u^{ma_n})}{(1-u)u^{b+1}}\right\}. \quad (14)$$

В дальнейшем везде будем считать, что все точки множества V_b равновероятны. Тогда значения функционала $f(x_1,\ldots,x_n) = \sum_{j=1}^{n} \mathrm{c}_j x_j$ — это случайная величина $\xi = \xi(a_1,\ldots,a_n,c_1,\ldots,c_n,b)$ с производящей функцией вероятностей:

$$P(z) = \frac{\Phi_b(z)}{\Phi_b(1)}.$$

Обозначим ее математическое ожидание $\mu(\xi)$. Математическое ожидание (первый момент) случайной величины определяется первой производной ее производящей функции вероятностей в точке $z = 1$.

Для определения первой производной функции $P(z)$, введем обозначение

$$\varphi(z,u) = \prod_{k=1}^{n}(1+z^{c_k}u^{a_k}+\cdots+z^{mc_k}u^{ma_k}).$$

Тогда производная выражения $\varphi(z,u)$ имеет вид:

$$\varphi'(z,u) = \sum_{k=1}^{n}\Bigg((c_kz^{c_k-1}u^{a_k} + 2c_kz^{2c_k-1}u^{2a_k}+\cdots+$$

$$+mc_kz^{mc_k-1}u^{ma_k})\prod_{\substack{i=1\\ i\neq k}}^{n}(1+z^{c_i}u^{a_i}+\cdots+z^{mc_i}u^{ma_i})\Bigg).$$

Выражая ее через $\varphi(z,u)$, получим:

$$\varphi'(z,u)=\sum_{k=1}^{n}\frac{c_k z^{c_k-1}u^{a_k}+2c_k z^{2c_k-1}u^{2a_k}+\cdots+mc_k z^{mc_k-1}}{1+z^{c_k}u^{a_k}+\cdots+z^{mc_k}u^{ma_k}}\varphi(z,u).$$

Отсюда выразим значение первой производной функции $\Phi_b(z)$.

$$\Phi_b'(z)=$$

$$=\underset{u}{coef}\left\{\sum_{k=1}^{n}\left(\frac{c_k z^{c_k-1}u^{a_k}+2c_k z^{2c_k-1}u^{2a_k}+\cdots+mc_k z^{mc_k-1}u^{ma_k}}{1+z^{c_k}u^{a_k}+\cdots+z^{mc_k}u^{ma_k}}\frac{\varphi(z,u)}{(1-u)u^{b+1}}\right)\right\}.$$

Подставив в это выражение $z=1$, получим:

$$\Phi_b'(1)=$$

$$=\underset{u}{coef}\left\{\sum_{k=1}^{n}\left(\frac{c_k u^{a_k}+2c_k u^{2a_k}+\cdots+mc_k u^{ma_k}}{1+u^{a_k}+\cdots+u^{ma_k}}\frac{\prod_{i=1}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1}}\right)\right\}. \quad (15)$$

Для каждой из *n* переменных введем *m+1* «сечений» множества допустимых решений задачи V_b следующим образом. Для переменной x_k *(1* $\leq k \leq$ *n)* сечение с номером *d (0* $\leq d \leq$ *m)* содержит все решения, удовлетворяющие условию: $\sum_{\substack{i=1\\ i\neq k}}^{n} a_i x_i \leq b-da_k, x_i \in \{0,1,\dots,m\}$. Эти решения соответствуют подмножеству решений задачи (5) с установленной в значение *d* переменной x_k. Обозначим это множество через V_b^{dk}.

Из следствия 2.1 получаем

$$V_b^{dk}=\underset{u}{coef}\left\{\frac{\prod_{\substack{i=1\\ i\neq k}}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1-da_k}}\right\}. \quad (16)$$

Теорема 2.1. Справедливо соотношение

$$\mu(\xi)=\frac{1}{|V_b|}\sum_{k=1}^{n}c_k\left(\left|V_b^{1k}\right|+2\left|V_b^{2k}\right|+\cdots+m\left|V_b^{mk}\right|\right). \quad (17)$$

Доказательство. Вспомним, что

$$P(z)=\frac{\Phi_b(z)}{\Phi_b(1)}.$$

И математическое ожидание случайной величины выражается как первая производная ее производящей функции $P(z)$ в точке $z=1$.

$$\mu(\xi) = P'(1) = \frac{\Phi_b'(1)}{\Phi_b(1)} \tag{18}$$

Разложим выражение из формулы (15) по первому множителю на m слагаемых

$$\Phi_b'(1) = \underset{u}{coef}\left\{\sum_{k=1}^{n}\left(\frac{c_k u^{a_k}}{1+u^{a_k}+\cdots+u^{ma_k}}\frac{\prod_{i=1}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1}}\right.\right.$$
$$+\frac{2c_k u^{2a_k}}{1+u^{a_k}+\cdots+u^{ma_k}}\frac{\prod_{i=1}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1}}+\cdots$$
$$\left.\left.+\frac{mc_k u^{ma_k}}{1+u^{a_k}+\cdots+u^{ma_k}}\frac{\prod_{i=1}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1}}\right)\right\}.$$

Теперь вынесем c_k за знак коэффициента и заметим, что слагаемые данного выражения содержат правые части выражений (16) для $d = 1, \dots, m$

$$\Phi_b'(1) = \sum_{k=1}^{n}\left(c_k \underset{u}{coef}\left\{\frac{\prod_{\substack{i=1\\ i\neq k}}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1-a_k}}\right\}\right.$$
$$+2c_k \underset{u}{coef}\left\{\frac{\prod_{\substack{i=1\\ i\neq k}}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1-2a_k}}\right\}+\cdots$$
$$\left.+mc_k \underset{u}{coef}\left\{\frac{\prod_{\substack{i=1\\ i\neq k}}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1-ma_k}}\right\}\right).$$

С учетом равенств (16), произведем замены коэффициентов их обозначениями и получим соотношение

$$\Phi_b'(1) = \sum_{k=1}^{n} c_k\left(\left|V_b^{1k}\right| + 2\left|V_b^{2k}\right| + \cdots + m\left|V_b^{mk}\right|\right).$$

С учетом (14) и (11) имеем

$$\Phi_b(1) = |V_b| = \underset{u}{coef}\left\{\frac{(1+u^{a_1}+\cdots+u^{ma_1})\dots(1+u^{a_n}+\cdots+u^{ma_n})}{(1-u)u^{b+1}}\right\}.$$

Подставляя найденные значения в (18), окончательно выводим искомое соотношение (17).

Данная формула может применяться при анализе и оценках как множества допустимых решений, так и значений функционала на оптимальных решениях, в частности, для оценок эффективности алгоритмов решения задачи (4), (5). В частности, среднее значение оптимизируемого функционала задачи может служить показателем качества решения при сравнении с результатами, полученными с применением эвристических или аппроксимационных алгоритмов. Если значения, полученные таким алгоритмом, существенно превышают среднее значение функционала, это говорит о том, что алгоритм обеспечивает решения, близкие к оптимальным. Кроме того, данная формула может быть применена для нахождения нижней оценки оптимального значения функционала задачи на подобласти допустимых значений переменной при использовании алгоритмов декомпозиции, например в методе ветвей и границ.

Выражение $|V_b|$ также можно представить через сумму $\left|V_b^{dk}\right|$, раскладывая по скобке, соответствующей переменной x_k

$$|V_b| = \underset{u}{coef}\left\{\frac{(1+u^{a_1}+\cdots+u^{ma_1})\ldots(1+u^{a_n}+\cdots+u^{ma_n})}{(1-u)u^{b+1}}\right\}.$$

Домножим и разделим на $(1+u^{a_k}+\cdots+u^{ma_k})$

$$|V_b| = \underset{u}{coef}\left\{\frac{1+u^{a_k}+\cdots+u^{ma_k}}{1+u^{a_k}+\cdots+u^{ma_k}}\frac{\prod_{i=1}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1}}\right\}.$$

Разложим теперь по числителю $(1+u^{a_k}+\cdots+u^{ma_k})$ на m слагаемых и заметим, что они содержат выражения (16) для $d=0,\ldots,m$

$$|V_b| = \underset{u}{coef}\left\{\frac{\prod_{\substack{i=1\\ i\neq k}}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1}}\right\} + \underset{u}{coef}\left\{\frac{\prod_{\substack{i=1\\ i\neq k}}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1-a_k}}\right\} +$$

$$+\cdots+\underset{u}{coef}\left\{\frac{\prod_{\substack{i=1\\ i\neq k}}^{n}(1+u^{a_i}+\cdots+u^{ma_i})}{(1-u)u^{b+1-ma_k}}\right\}.$$

Заменяя коэффициенты их обозначениями из (16), получим

$$|V_b| = \left|V_b^{0k}\right| + \left|V_b^{1k}\right| + \cdots + \left|V_b^{mk}\right|. \qquad (19)$$

Эта формула позволяет сократить количество рассчитываемых значений в формуле (17). Для этого достаточно подставить в (17) значение $|V_b|$, определенное по формуле (19) для какой-нибудь одной переменной x_j, например, наименьшего значения a_j. Тогда

$$\mu(\xi) = \frac{1}{|V_b|}\sum_{k=1}^{n} c_k\left(\left|V_b^{1k}\right| + 2\left|V_b^{2k}\right| + \cdots + m\left|V_b^{mk}\right|\right) =$$

$$= \frac{1}{\left|V_b^{0j}\right| + \left|V_b^{1j}\right| + \cdots + \left|V_b^{mj}\right|}\sum_{k=1}^{n} c_k\left(\left|V_b^{1k}\right| + 2\left|V_b^{2k}\right| + \cdots + m\left|V_b^{mk}\right|\right). \qquad (20)$$

Применение формулы (20) и связанных с ней выражений позволяет проводить эффективные оценки мощности множества допустимых решений задачи о рюкзаке, что имеет фундаментальное значение для анализа вычислительной сложности алгоритмов и исследования структурных свойств пространства решений задачи.

В контексте оценки вычислительной сложности задачи о рюкзаке особый интерес представляют аналитические выражения для среднего числа допустимых решений при фиксированной размерности задачи. Такие оценки позволяют формализовать связь между структурой множества решений и параметрами входных данных, что важно для анализа поведения алгоритмов и прогнозирования их трудоемкости. Определение среднего числа решений дает возможность оценить степень «перегруженности» пространства допустимых конфигураций, установить границы применимости переборных и аппроксимационных методов, а также сравнивать вычислительную сложность различных классов экземпляров задачи. Полученные оценки служат основой для классификации задач по степени сложности и позволяют уточнить представления о распределении вычислительно трудных и вычислительно простых экземпляров в пространстве входных параметров.

В выражении (11) была определена формула для нахождения числа решений в конкретной задаче о рюкзаке. Определим теперь среднее число

решений на некотором множестве задач о рюкзаке с фиксированными параметрами.

Обозначим $\left|\bar{V}_p\right|$ – среднее число решений набора задач об ограниченном рюкзаке (4), (5) с коэффициентами весов $a_i, i = 1, \dots, n$, не превосходящими некоторого заранее заданного значения *p*. Значение этой величины выражается по формуле

$$\left|\bar{V}_p\right| = \frac{1}{(p+1)^n} \sum_{0 \le a_i \le p, i=1,\dots,n} \left|V_b(a_1, \dots, a_n)\right|. \qquad (21)$$

Рассмотрим вопрос о среднем числе допустимых решений задач о рюкзаке при некоторых значениях числа копий предметов m. Пусть значение b и размерность задачи n фиксированы, при этом компоненты вектора весов $(a_1, \dots, a_n)$ принимают значения от 0 до *b*. Формула для вычисления среднего числа решений по всем таким задачам в частном случае, когда $m = 1$, получена и доказана в работе [52]. Далее приводится формулировка соответствующей теоремы, которая будет обобщена на случай, когда переменные принимают значения из множества $x \in \{0,1,2\}^n$.

Теорема 2.2. При $x \in \{0,1\}^n$ справедлива формула:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \sum_{k=0}^{n} C_n^k C_b^{n-k} (b+2)^k. \qquad (22)$$

Полученная аналитическая формула позволяет осуществлять обоснованный выбор алгоритмического подхода к решению задачи на основе количественных оценок мощности пространства решений. Данный результат может быть применен для оценки априорной вероятности успешного решения задачи, а также для анализа вычислительной сложности алгоритмов полного перебора. Эта формула будет использована в главе 3 для сравнительного анализа сложности решения различных классов экземпляров задачи о рюкзаке.

Рассмотрим теперь вопрос о среднем значении мощности множества допустимых решений в более общем случае. Следующая производящая функция

выражает число решений каждой задачи размерности n с компонентами вектора весов $(a_1, \dots, a_n)$, принимающими значения в диапазоне от 0 до p.

$$R_p(z_1, \dots, z_n) = \sum_{0 \le a_i \le p, i=1,\dots,n} z_1^{a_1}, \dots, z_n^{a_n} |V_b(a_1, \dots, a_n)| . \quad (23)$$

Теорема 2.3. Справедлива формула

$$R_p(z_1, \dots, z_n) = \underset{u}{coef} \left\{ \prod_{k=1}^{n} \frac{\frac{1-z_k^{p+1}}{1-z_k} + \frac{1-(z_k u)^{p+1}}{1-z_k u} + \dots + \frac{1-(z_k u^m)^{p+1}}{1-z_k u^m}}{u^b (1-u)} \right\} \quad (24)$$

Доказательство. Подставим в (23) выражение для числа решений из формулы (11)

$$R_p(z_1, \dots, z_n) =$$

$$= \sum_{0 \le a_i \le p, i=1,\dots,n} z_1^{a_1}, \dots, z_n^{a_n} \underset{u}{coef} \left\{ \frac{(1 + u^{a_1} + \dots + u^{ma_1}) \dots (1 + u^{a_n} + \dots + u^{ma_n})}{(1-u)u^{b+1}} \right\}.$$

Объединим выражения, в которых суммирование производится по одинаковому компоненту a_i:

$$R_p(z_1, \dots, z_n) =$$

$$= \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \left(\sum_{a_1=0}^{p} z_1^{a_1} (1 + u^{a_1} + \dots + u^{ma_1}) \dots \sum_{a_n=0}^{p} z_n^{a_n} \cdot \right. \right.$$

$$\left. \left. \cdot (1 + u^{a_n} + \dots + u^{ma_n}) \right) \right\}.$$

Теперь заметим, что каждое выражение под знаком суммы можно разложить в $p+1$ сумму геометрической прогрессии:

$$\sum_{a_k=0}^{p} z_k^{a_k} (1 + u^{a_k} + \dots + u^{ma_k}) = \sum_{a_k=0}^{p} z_k^{a_k} + \sum_{a_k=0}^{p} z_k^{a_k} u^{a_k} + \dots + \sum_{a_k=0}^{p} z_k^{a_k} u^{ma_k} =$$

$$= \frac{1-z_k^{p+1}}{1-z_k} + \frac{1-(z_k u)^{p+1}}{1-z_k u} + \dots + \frac{1-(z_k u^m)^{p+1}}{1-z_k u^m}.$$

Подставляя полученное выражение в исходную формулу, получим искомое выражение.

Подставляя во все аргументы левой части формулы (24) значение z, получаем производящую функцию, выражающую общее число решений задач с одинаковой суммой коэффициентов:

$$R_p(z) = \underset{u}{coef}\left\{\left(\frac{1-z^{p+1}}{1-z} + \frac{1-(zu)^{p+1}}{1-zu} + \cdots + \frac{1-(zu^m)^{p+1}}{1-zu^m}\right)^n \middle/ u^b(1-u)\right\}.$$

Данная производящая функция может быть адаптирована для решения задач, связанных с использованием специфичных комбинаторных моделей, отражающих особенности конкретной области.

Теперь перейдем к следующему случаю.

Теорема 2.4. При $x \in \{0,1,2\}^n$ справедлива формула:

$$|\bar{V}_b| = \sum_{k=0}^{n} C_n^k (b+1)^{k-n} \sum_{t=0}^{n-k} C_{n-k}^t 2^t \left(C_{\frac{n-k+t+b-1}{2}}^{n-k} [n-k+t+b \pmod 2 = 1] + \right.$$

$$\left. + C_{\frac{n-k+t+b}{2}}^{n-k} [n-k+t+b \pmod 2 = 0] \right). \qquad (25)$$

Здесь $[P]$ – скобка Айверсона, равная 1, если условие P выполняется, и 0 в противном случае.

Доказательство. Подставляя значение из формулы (11) при *m=2* в выражение (21), получаем

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef}\left\{\frac{\sum_{a_1=0}^{b}(1+u^{a_1}+u^{2a_1}) \ldots \sum_{a_n=0}^{b}(1+u^{a_n}+u^{2a_n})}{(1-u)u^{b+1}}\right\}.$$

Аналогично предыдущей теореме можем разложить суммы под знаком коэффициента в две суммы геометрических прогрессий:

$$\sum_{a_i=0}^{b}(1+u^{a_i}+u^{2a_i}) = b+3+\sum_{a_i=1}^{b}u^{a_i}+\sum_{a_i=1}^{b}u^{2a_i} =$$

$$= b+3+\frac{(1-u^b)u}{1-u}+\frac{(1-u^{2b})u^2}{1-u^2}.$$

Подставляя полученное выражение вместо сумм по a_i для каждого $i = 1, \dots, n$, получим:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{\left(b + 3 + \frac{(1-u^b)u}{1-u} \left(1 + \frac{(1+u^b)u}{1+u} \right) \right)^n}{(1-u)u^{b+1}} \right\}.$$

Разложим числитель по формуле бинома Ньютона:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \sum_{k=0}^{n} C_n^k (b+3)^k \left(\frac{u(1-u^b)}{1-u} \right)^{n-k} \cdot \right.$$

$$\left. \cdot \left(1 + \frac{(1+u^b)u}{1+u} \right)^{n-k} \right\}.$$

Получившуюся сумму разложим на слагаемые с $k < n$ и $k = n$:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \sum_{k=0}^{n-1} C_n^k (b+3)^k \left(\frac{u(1-u^b)}{1-u} \right)^{n-k} \cdot \right.$$

$$\left. \cdot \left(1 + \frac{(1+u^b)u}{1+u} \right)^{n-k} \right\} + \frac{(b+3)^n}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \right\}. \qquad (26)$$

Теперь разложим множитель $(1-u^b)^{n-k}$ из числителя первого слагаемого по формуле бинома Ньютона:

$$(1-u^b)^{n-k} = 1 - C_{n-k}^1 u^b + C_{n-k}^2 u^{2b} + \cdots + (-1)^{n-k} C_{n-k}^{n-k} u^{(n-k)b} =$$

$$= \sum_{i=0}^{n-k} (-1)^i C_{n-k}^i u^{ib}. \qquad (27)$$

Подставляя в формулу (26) разложение (27), получаем

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \sum_{k=0}^{n-1} C_n^k (b+3)^k \left(\frac{u}{1-u} \right)^{n-k} \cdot \right.$$

$$\left. \cdot \left(1 + \frac{(1+u^b)u}{1+u} \right)^{n-k} \sum_{i=0}^{n-k} (-1)^i C_{n-k}^i u^{ib} \right\}$$

$$+ \frac{(b+3)^n}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \right\}.$$

Из свойств коэффициента следует, что данное выражение принимает нулевое значение, когда степень u в числителе больше или равна степени u с положительным знаком в знаменателе. Поскольку в первом слагаемом $k < n$, данное выражение обращается в нуль для всех $i > 0$. Таким образом, получаем

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \sum_{k=0}^{n-1} C_n^k (b+3)^k \left(\frac{u}{1-u}\right)^{n-k} \cdot \right.$$

$$\left. \cdot \left(1 + \frac{(1+u^b)u}{1+u}\right)^{n-k} \right\} + \frac{(b+3)^n}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \right\}.$$

Разложим множитель $\left(1 + \frac{(1+u^b)u}{1+u}\right)^{n-k}$ в первом слагаемом по формуле бинома Ньютона:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \sum_{k=0}^{n-1} C_n^k (b+3)^k \left(\frac{u}{1-u}\right)^{n-k} \cdot \right.$$

$$\left. \cdot \sum_{t=0}^{n-k} C_{n-k}^t \left(\frac{(1+u^b)u}{1+u}\right)^t \right\} + \frac{(b+3)^n}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \right\}. \qquad (28)$$

Разложим теперь первый коэффициент по множителю $(1+u^b)$ аналогично
(24):

$$(1+u^b)^{n-k} = 1 + C_{n-k}^1 u^b + C_{n-k}^2 u^{2b} + \cdots + C_{n-k}^{n-k} u^{(n-k)b} = \sum_{i=0}^{n-k} C_{n-k}^i u^{ib}.$$

Подставляя это разложение в выражение (28) и проводя аналогичные рассуждения, получаем:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \sum_{k=0}^{n-1} C_n^k (b+3)^k \left(\frac{u}{1-u}\right)^{n-k} \cdot \right.$$

$$\left. \cdot \sum_{t=0}^{n-k} C_{n-k}^t \left(\frac{u}{1+u}\right)^t \right\} + \frac{(b+3)^n}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \right\}.$$

Преобразуем обратно в бином последнюю сумму в первом слагаемом:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \sum_{k=0}^{n-1} C_n^k (b+3)^k \left(\frac{u}{1-u}\right)^{n-k} \cdot \right.$$

$$\left. \cdot \left(\frac{u}{1+u} + 1\right)^{n-k} \right\} + \frac{(b+3)^n}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \right\}.$$

Заметим, что $C_n^n (b+3)^n \left(\frac{u}{1-u}\right)^{n-n} \left(\frac{u}{1+u} + 1\right)^{n-n} = 1$, и занесем второе слагаемое под знак суммы первого:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \sum_{k=0}^{n} C_n^k (b+3)^k \left(\frac{u}{1-u}\right)^{n-k} \left(\frac{u}{1+u} + 1\right)^{n-k} \right\}.$$

И снова соберем получившуюся сумму в бином:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \left(b + 3 + \frac{2u^2 + u}{1-u^2} \right)^n \right\}.$$

Теперь разложим по формуле бинома Ньютона, оставляя $+2$ во втором слагаемом:

$$|\bar{V}_b| = \frac{1}{(b+1)^n} \underset{u}{coef} \left\{ \frac{1}{(1-u)u^{b+1}} \sum_{k=0}^{n} C_n^k (b+1)^k \left(\frac{u+2}{1-u^2}\right)^{n-k} \right\}.$$

Вынесем множители, не зависящие от u, за знак коэффициента, разложим выражение по последнему множителю и сгруппируем получившиеся множители:

$$|\bar{V}_b| = \sum_{k=0}^{n} C_n^k (b+1)^{k-n} \underset{u}{coef} \left\{ \sum_{t=0}^{n-k} C_{n-k}^t 2^t u^{n-k-t-b-1} (1-u^2)^{k-n-1} (1+u) \right\}.$$

Разложив последнее выражение по множителю $(1+u)$ на два слагаемых и преобразовав получившиеся выражения в биномиальные коэффициенты в соответствии с правилом (3) метода коэффициентов для u^2, получим:

$$|\bar{V}_b| = \sum_{k=0}^{n} C_n^k (b+1)^{k-n} \sum_{t=0}^{n-k} C_{n-k}^t 2^t \cdot$$

$$\cdot \left(C_{k-n-1}^{\frac{k-n+t+b-1}{2}} (-1)^{\frac{k-n+t+b-1}{2}} [n-k+t+b \pmod 2 = 1] + \right.$$

$$+ C_{k-n-1}^{\frac{k-n+t+b}{2}} (-1)^{\frac{k-n+t+b}{2}} [n-k+t+b \pmod 2 = 0]\Big).$$

Наконец, преобразуя биномиальные коэффициенты по правилу $(-1)^{n-m} C_{-(m+1)}^{n-m} = C_n^m$ [53], получим искомое выражение (25).

Полученные формулы могут быть полезны для выбора оптимального подхода к решению для многозначных постановок задачи о рюкзаке, определения вероятности их успешного решения, а также для исследования свойств задач и оптимизации процесса поиска их решений.

2.2 Линейные формы с высокой плотностью

Рассмотрим частный случай задачи $\sum_{i=1}^{n} a_i x_i = b$, при котором частичные суммы компонентов рюкзачного вектора принимают все целочисленные значения от 0 до $\sum_{i=1}^{n} a_i$.

Обозначим $L(x_1, \dots, x_n) = \sum_{i=1}^{n} a_i x_i$ линейную форму, состоящую из компонентов вектора весов задачи о рюкзаке. Накладывая определенные условия на вид этой формы, можно получить различные частные случаи общей задачи. В задаче о 0-1 рюкзаке переменные x_i являются булевыми. При отсутствии неопределенности, в общем виде будем также обозначать эту линейную форму $L(x)$. Множество значений линейной формы $L(x_1, \dots, x_n)$ будем обозначать $L^*(x_1, \dots, x_n)$. Уравнение $L(x) = b$ разрешимо тогда и только тогда, когда $b \in L^*(x)$.

Ранее определение линейной формы с булевыми переменными $f(x_1, \dots, x_n) = \sum_{j=1}^{n} c_j x_j$, принимающей все значения из интервала $[0, \sum_{j=1}^{n} c_j]$, было дано в работе [54].

Без ограничения общности далее будем считать, что $a_1 \le a_2 \le \dots \le a_n$ и $x \in \{0,1\}$.

Определение 2.1. Линейная форма $L(x_1, \dots, x_n) = \sum_{i=1}^{n} a_i x_i$ называется **сюръективной**, если $\forall\, b \in \{0, 1, 2, \dots, \sum_{i=1}^{n} a_i\}\ b \in L^*(x)$. Далее будем обозначать $L^*(x) = [0, \sum_{i=1}^{n} a_i]$.

Утверждение 2.1. Справедливо соотношение $|L^*(x_1, \dots, x_n)| \leq min\{2^n, \sum_{i=1}^n a_i\}$.

Важность сюръективных линейных форм объясняется тем, что проверка на разрешимость системы булевых уравнений, левые части которых образованы сюръективными линейными формами, является тривиальной. Достаточно для каждого уравнения проверить выполнение $b \leq \sum_{i=1}^n a_i$.

В следующей теореме выведены необходимые и достаточные условия сюръективности линейной формы.

Теорема 2.5. Форма $L(x_1, \dots, x_n) = \sum_{i=1}^n a_i x_i$ является сюръективной тогда и только тогда, когда

$$a_1 = 1, \text{и } a_k \leq \sum_{i=1}^{k-1} a_i + 1 \; \forall \; k = 2, \dots, n. \tag{29}$$

Доказательство. Достаточность. Пусть $a_1 = 1$ и $a_k \leq \sum_{i=1}^{k-1} a_i + 1 \; \forall \; k = 2, \dots, n$. Покажем, что $L(x_1, \dots, x_n)$ сюръективна индукцией по числу переменных. В качестве базы возьмем линейную форму $L(x_1, x_2)$. Положим $a_1 = 1$, если $a_2 = 1$, то $L^* = \{0,1,2\}$, $a_2 = 2$, то $L^* = \{0,1,2,3\}$. Пусть теперь $L^*(x_1, \dots, x_{k-1}) = [0, \sum_{i=1}^{k-1} a_i]$ и $a_k \leq \sum_{i=1}^{k-1} a_i + 1$. Форма $L(x_1, \dots, x_n)$ принимает значения $\left\{\sum_{i=1}^{k-1} a_i x_i + 0 \cdot a_k, \; x_i \in \{0,1\}, \; i = 1, \dots, k\right\} = [0, \sum_{i=1}^{k-1} a_i]$ и $\left\{\sum_{i=1}^{k-1} a_i x_i + 1 \cdot a_k, \; x_i \in \{0,1\}, \; i = 1, \dots, k\right\} = \left[a_k, a_k + \sum_{i=1}^{k-1} a_i\right]$. Поскольку $a_k \leq \sum_{i=1}^{k-1} a_i + 1$, то отрезки $[0, \sum_{i=1}^{k-1} a_i]$ и $[a_k, \sum_{i=1}^{k-1} a_i + a_k]$ пересекаются. Следовательно, $L(x_1, \dots, x_k)$ сюръективна с областью значений $[0, \sum_{i=1}^{k} a_i]$.

Необходимость. Пусть $L(x_1, \dots, x_n)$ сюръективна, покажем, что $a_k \leq \sum_{i=1}^{k-1} a_i + 1 \; \forall \; k = 1, \dots, n$. Если $a_1 > 1$ то $1 \notin L^*$ и сюръективность нарушена. Пусть теперь условие выполнено для всех $i < k$ и $a_k > \sum_{i=1}^{k-1} a_i + 1$ для некоторого k. Тогда $\sum_{i=1}^{k-1} a_i + 1 \notin L^*$, поскольку $\sum_{i=1}^{k-1} a_i < \sum_{i=1}^{k-1} a_i + 1$, а $\forall t \geq k \; a_t > \sum_{i=1}^{k-1} a_i + 1$.

Аналогичные условия можно вывести и для общего случая.

Следствие 2.2. Пусть $a_1 \le a_2 \le \cdots \le a_n$ и $x \in \{0,1,\dots,p\}$. Форма $L(x_1,\dots,x_n) = \sum_{i=1}^{n} a_i x_i$ сюръективна тогда и только тогда, когда $a_1 = 1$, $a_k \le p\sum_{i=1}^{k-1} a_i + 1 \;\forall\, k = 1,\dots,n$.

Доказательство. Аналогично доказательству теоремы 2.5.

Установленное в теореме необходимое и достаточное условие сюръективности линейной формы демонстрирует конструктивный характер соответствующего класса комбинаторных объектов. Процедура построения сюръективных форм обладает сравнительной простотой, аналогичной алгоритму генерации сверхрастущих последовательностей. Данное свойство открывает возможности для конструирования экземпляров задачи о рюкзаке с заданными вычислительными характеристиками, что представляет интерес для исследования границ между простыми и трудными для решения случаями задачи.

Пример 1. Примерами сюръективных линейных форм при любом *p>0* являются:

1. $f(x_1,\dots,x_n) = \sum_{j=1}^{n} 2^{j-1} x_j$;
2. $f(x_1,\dots,x_n) = \sum_{j=1}^{n} j x_j$;

А форма $2x_1 + 3x_2 + 4x_3$ не является сюръективной при любом *p>0*, поскольку она не принимает значение 1.

Условия (29) будем также называть условиями сюръективности линейной формы. Из них теперь можно определить максимальные значения компонент сюръективной формы.

Утверждение 2.2. Форма $L(x_1,\dots,x_n) = \sum_{i=1}^{n} 2^{i-1} x_i$, где $x \in \{0,1\}$, является сюръективной линейной формой с максимальными компонентами.

Доказательство. Возьмем $a_k = \sum_{i=1}^{k-1} a_i + 1 \;\forall\, k = 1,\dots,n$. Легко проверить, что $a_k = 2^{k-1}$.

Из данного утверждения можно определить минимальную плотность сюръективного рюкзачного вектора.

Следствие 2.3. Плотность сюръективного рюкзачного вектора составляет $d(A) \geq \frac{n}{n-1} > 1$.

Важным структурным свойством рюкзачных векторов является инъективность, которая гарантирует, что каждое достижимое значение суммы соответствует единственному решению, что существенно влияет на сложность алгоритмов решения и проверки решений.

Определение 2.2. Рюкзачный вектор $A = (a_1, \ldots, a_n)$ называется инъективным, если $\forall\, A', A'' \subset A, A' \neq A''$ выполняется $\sum_{a' \in A'} a' \neq \sum_{a'' \in A''} a''$.

Следующее утверждение следует из теоремы 2.5.

Утверждение 2.3. За исключением случая $L(x_1, \ldots, x_n) = \sum_{i=1}^{n} 2^{i-1} x_i$, ни одна сюръективная линейная форма не является инъективной.

Таким образом, системы на основе сюръективных рюкзачных векторов обладают повышенной устойчивостью к алгоритмам, использующим низкую плотность вектора. Вместе с тем, свойство сюръективности приводит к неинъективности отображения, что обуславливает множественность решений задачи. Для практического применения данного подхода необходимо обеспечить достаточную мощность множества сюръективных форм, чтобы исключить возможность решения полным перебором по пространству возможных конфигураций.

Утверждение 2.4. Обозначим общее число сюръективных линейных форм от n переменных через $L_c(n)$. Оно может быть посчитано по формуле: $L_c(n) = \sum_{a_2=1}^{2} \sum_{a_3=a_2}^{a_2+2} \cdots \sum_{a_n=a_{n-1}}^{a_2+\cdots+a_{n-1}+2} 1$.

Поскольку данное выражение определяет вложенное суммирование, в котором пределы каждого суммирования зависят от предыдущих индексов, оно не может быть легко преобразовано в выражение замкнутой формы. Однако, для него могут быть найдены верхняя и нижняя оценки.

Следствие 2.4. $n! < L_c(n) < 2^{n \cdot \frac{n-1}{2}}$.

Доказательство. В качестве верхней границы возьмем компоненты максимальной сюръективной линейной формы. Тогда $L_c(n) =$

$\sum_{a_2=1}^{2}\sum_{a_3=a_2}^{a_2+2}\cdots\sum_{a_n=a_{n-1}}^{a_2+\cdots+a_{n-1}+2}1 < \sum_{a_2=1}^{2}\sum_{a_3=1}^{a_2+2}\cdots\sum_{a_n=1}^{a_2+\cdots+a_{n-1}+2}1 <$

$\sum_{a_2=1}^{2}\sum_{a_3=1}^{4}\cdots\sum_{a_n=1}^{2^{n-1}}1 = \prod_{i=0}^{n-1}2^i = 2^{\sum_{i=0}^{n-1}i} = 2^{n\cdot\frac{n-1}{2}}$. Поскольку при любых значениях $a_i, i = 1,\dots,k$ для $a_k \in [a_{k-1}; a_{k-1}+k-1]$ выполняется $a_k \le a_{k-1} + k - 1 \le \sum_{i=1}^{k-1}a_i + 1$, то $L_c(n) \ge \sum_{a_2=1}^{2}\sum_{a_3=a_2}^{a_2+2}\cdots\sum_{a_n=a_{n-1}}^{a_{n-1}+n-1}1 = \sum_{i=1}^{2}\sum_{i=1}^{3}\cdots\sum_{i=1}^{n}1 = n!$

Полученная нижняя оценка мощности множества сюръективных линейных форм свидетельствует об экспоненциальном росте их количества с увеличением размерности задачи. При достаточно больших значениях параметра *n* данный класс комбинаторных объектов образует обширное семейство, что позволяет рассматривать его как самостоятельное значимое множество в пространстве всех экземпляров задачи о рюкзаке, а не как частный случай.

Наиболее важным свойством сюръективных форм с точки зрения вычислительной сложности является возможность нахождения всех решений соответствующей задачи за время, линейно зависящее от размерности задачи и числа решений. Данное свойство выделяет сюръективные формы в качестве класса задач о рюкзаке, для которых существует эффективный алгоритм полного перебора решений.

Теорема 2.6. Если форма $L(x_1,\dots,x_n) = \sum_{i=1}^{n}a_ix_i$ сюръективна, $b \in L^*(x_1,\dots,x_n)$, то все решения задачи $\sum_{i=1}^{n}a_ix_i = b$ можно найти за время $O(ns)$, где *s* – число этих решений.

Доказательство. Построим рекурсивную процедуру поиска решений. Проходя индукцией по переменным начиная с конца, покажем, что всякий раз мы переходим к решению разрешимого равенства меньшей размерности. Будем считать, что $\sum_{i=1}^{n}a_i \ge b$, иначе у системы нет решений. База индукции: по условию форма сюръективна, поэтому $\sum_{i=1}^{n}a_ix_i = b$ разрешима. Шаг индукции: пусть ранее были найдены значения $x_n,\dots,x_{k+1}$ и $\sum_{i=1}^{k}a_i \ge b$. Найдем допустимые значения для равенства $\sum_{i=1}^{k}a_ix_i = b$. Поскольку форма $\sum_{i=1}^{k}a_ix_i$ сюръективна, возможны три непересекающихся случая:

1. Если $\sum_{i=1}^{k-1} a_i < b$, то единственным допустимым значением будет $x_k = 1$ и на следующий шаг переходим с $b = b - a_k$. По условию $\sum_{i=1}^{k} a_i \geq b$, следовательно $\sum_{i=1}^{k-1} a_i \geq b - a_k$, поэтому равенство $\sum_{i=1}^{k-1} a_i x_i = b - a_k$ разрешимо. (Нетрудно заметить, что сюда же попадут и все случаи $b \geq a_k = \sum_{i=1}^{k} a_i + 1$).
2. Если $a_k > b$, то единственным допустимым значением будет $x_k = 0$ и на следующем шаге ищем решения равенства $\sum_{i=1}^{k-1} a_i x_i = b$. По условию $\sum_{i=1}^{k} a_i + 1 \geq a_k > b$, следовательно $\sum_{i=1}^{k-1} a_i \geq b$, поэтому система $\sum_{i=1}^{k-1} a_i x_i = b$ разрешима.
3. Если $\sum_{i=1}^{k-1} a_i \geq b \geq a_k$, то допустимыми для x_k будут два значения. Для $x_k = 1$ далее ищем решения равенства $\sum_{i=1}^{k-1} a_i x_i = b - a_k$. Для $x_k = 0$ ищем решения равенства $\sum_{i=1}^{k-1} a_i x_i = b$. Из условий видно, что оба этих равенства разрешимы.

Таким образом каждое решение исходной задачи гарантированно обнаруживается за n шагов. Причем на каждом шаге либо выбирается единственное допустимое значение переменной x_i, либо происходит ветвление алгоритма по двум значениям переменной, которые соответствуют различным решениям исходной задачи. А поскольку в конце каждого шага получается разрешимое равенство, то общее число ветвей в алгоритме равно числу решений изначальной задачи s.

Для анализа устойчивости структурных свойств сюръективных форм к алгебраическим преобразованиям представляет интерес исследование их поведения при модульных отображениях. С точки зрения комбинаторного анализа, применение модульного умножения изменяет арифметическую структуру вектора коэффициентов, но не нарушает общие закономерности структуры множества решений в задаче о рюкзаке.

Определение 2.3. Пусть дан вектор целых чисел $A = (a_1, a_2, \dots, a_n)$, целое число $M > \sum_{i=1}^{n} a_i$, и натуральное $t < M$, такое что $\gcd(t, M) = 1$. Тогда говорят,

что вектор $B = (b_1, b_2, \dots, b_n)$, где $b_i = (ta_i mod M)$, $i = 1, \dots, n$ получен из A **сильным модульным умножением** относительно пары (M, t).

Условие $(t, M) = 1$ гарантирует существование обратного элемента $t^{-1} = u$, удовлетворяющего соотношению $tu \equiv 1(\mathrm{mod}\, M), 1 \leq u < M$, что обеспечивает возможность восстановления исходного вектора A из B обратным модульным умножением относительно того же модуля M и множителя u.

Такое преобразование играет важную роль при исследовании трудных экземпляров задачи о рюкзаке. В частности, сильное модульное умножение нарушает свойства упорядоченности и сюръективности, характерные для исходных простых форм, в результате чего задача, ранее допускавшая эффективное решение, становится вычислительно сложной. При этом структура множества допустимых значений сохраняется, что делает модульное преобразование удобным инструментом для построения задач с управляемой сложностью [55].

Пусть форма $B(x_1, \dots, x_n)$ получена из формы $A(x_1, \dots, x_n)$ посредством сильного модульного умножения $b_i \equiv ua_i(\mathrm{mod} M), (u, M) = 1$, и множитель u рассматривается как случайная величина, равномерно распределенная по множеству обратимых по модулю M элементов, т.е. по мультипликативной группе $(\mathbb{Z}/M\mathbb{Z})^{\times}$.

Поскольку умножение на обратимый элемент по модулю M является биекцией, отображение $\pi_u: x \to ux(\mathrm{mod} M)$ задает перестановку множества $\{1,2, \dots, M-1\}$. При этом семейство перестановок $\{\pi_u\}$ не является равномерным по всей симметрической группе, и потому при фиксированном u коэффициенты формы B в общем случае не являются ни независимыми, ни равномерно распределенными.

Тем не менее, усреднение по u позволяет получить однородные статистические характеристики образов фиксированных коэффициентов. Рассмотрим вероятность и математическое ожидание, взятые по равномерному выбору $u \in (\mathbb{Z}/M\mathbb{Z})^{\times}$.

В общем случае, при составном модуле M, отображение $u \to ua_i (\mathrm{mod} M)$, $u \in (\mathbb{Z}/M\mathbb{Z})^{\times}$, принимает значения, кратные $d_i = \gcd(a_i, M)$. Образ данного отображения представляет собой орбиту элемента a_i относительно действия мультипликативной группы и имеет мощность $\varphi(M_i)$, где $M_i = M/d_i$.

Для любого $t \in \{1, \dots, M-1\}$ вероятность события $b_i \leq t$, рассматриваемая по равномерному выбору $u \in (\mathbb{Z}/M\mathbb{Z})^{\times}$, допускает точное представление

$$\mathrm{P}_u(b_i \leq t) = \frac{\#\{v \leq \frac{t}{d_i} : \gcd(v, M_i) = 1\}}{\varphi(M_i)}.$$

Обозначим

$$S(X, m) = \#\{v \leq X : \gcd(v, m) = 1\}.$$

И

$$\mathbf{1}_{\gcd(v,m)=1} = \sum_{d \mid \gcd(v,m)} \mu(d),$$

где μ – функция Мёбиуса.

Отсюда, выполняя перестановку сумм, получим

$$S(X, m) = \sum_{v \leq X} 1 \sum_{d \mid \gcd(v,m)} \mu(d) = \sum_{d \mid m} \mu(d) \sum_{\substack{v \leq X \\ d \mid v}} 1 = \sum_{d \mid m} \mu(d) \left\lfloor \frac{X}{d} \right\rfloor.$$

Раскладывая $\left\lfloor \frac{X}{d} \right\rfloor = \frac{X}{d} + O(1)$, получаем

$$S(X, m) = X \sum_{d \mid m} \frac{\mu(d)}{d} + O\left(\sum_{d \mid m} 1 \right).$$

Учитывая $\sum_{d \mid m} \frac{\mu(d)}{d} = \frac{\varphi(m)}{m}$ и обозначая $\sum_{d \mid m} 1 = \tau(m)$, имеем

$$S(X, m) = X \frac{\varphi(m)}{m} + O(\tau(m)).$$

Подставляя полученное значение в выражение для вероятности с учетом $d_i M_i = M$, получаем

$$P_u(b_i \leq t) = \frac{t}{M} + O\left(\frac{\tau(M_i)}{\varphi(M_i)} \right).$$

Таким образом, для каждого фиксированного индекса i распределение величины b_i по случайному выбору множителя u обладает тем же линейным главным членом t/M, что и равномерное распределение на $\{1, \dots, M-1\}$, а отклонение от него определяется величиной $\tau(M_i)/\varphi(M_i)$, зависящей лишь от арифметической структуры модуля.

Если $(a_i, M) = 1$, то отображение $u \to ua_i (\mathrm{mod} M)$ является биекцией группы $(\mathbb{Z}/M\mathbb{Z})^\times$ на себя. В этом случае значения b_i равномерно распределены по множеству обратимых вычетов, и для любого фиксированного i и любого $t \in \{1, \dots, M-1\}$ имеет место оценка

$$\mathrm{P}_u(b_i \le t) = \frac{t}{M} + O\left(\frac{1}{\varphi(M)}\right).$$

Рассмотрим случайную величину $N(t) = \#\{i : b_i \le t\}$, равную числу коэффициентов формы B, не превосходящих t. По линейности математического ожидания получаем

$$\mathrm{E}_u[N(t)] = \sum_{i=1}^{n} \mathrm{P}_u\,(b_i \le t) = \frac{nt}{M} + O\left(\sum_{i=1}^{n} \frac{\tau(M_i)}{\varphi(M_i)}\right).$$

Для сюръективных форм с модулем M, соизмеримым с суммой коэффициентов, типичные значения a_i существенно меньше M, вследствие чего для подавляющего большинства индексов i выражение $\frac{\tau(M_i)}{\varphi(M_i)}$ существенно меньше $\frac{t}{M}$, и ошибка в указанной оценке оказывается пренебрежимо малой. В этом режиме усредненные по u порядковые характеристики коэффициентов формы B совпадают по главному члену с соответствующими характеристиками случайной подвыборки из множества $\{1, \dots, M-1\}$, что позволяет применять вероятностную модель случайной выборки при вычислении математических ожиданий порядковых статистик.

Таким образом, хотя при фиксированном u коэффициенты формы B не образуют равномерной случайной выборки, их усредненные по u порядковые характеристики по главному члену совпадают с характеристиками случайной подвыборки без возвращения. В этом смысле при анализе типичного поведения

упорядоченных коэффициентов формы B допустимо использовать вероятностную модель случайной подвыборки.

Утверждение 2.5. Пусть коэффициенты формы $B(x_1, \dots, x_n)$, полученной из формы $A(x_1, \dots, x_n)$ посредством сильного модульного умножения $b_i = ua_i(\mathrm{mod}M), (u, M) = 1$, рассматриваются как случайная выборка n элементов из множества $\{1,2, \dots, M-1\}$.

Обозначим упорядоченные компоненты формы $B(x_1, \dots, x_n)$ через $b^{(1)} < b^{(2)} < \cdots < b^{(n)}$. Тогда математическое ожидание компоненты $b^{(k)}$ равно:

$$E\left[b^{(k)}\right] = \frac{Mk}{n+1}, \qquad k = 1, \dots, n.$$

Доказательство. В рамках принятой вероятностной модели значения коэффициентов формы B образуют случайную выборку из множества $\{1,2, \dots, M-1\}$. Тогда $b^{(k)}$ является k-й порядковой статистикой данной выборки. Далее доказательство стандартным образом следует из комбинаторного выражения для распределения порядковых статистик:

$$P\left(b_{(k)} = x\right) = \frac{C_{x-1}^{k-1} C_{M-1-x}^{n-k}}{C_{M-1}^{n}}$$

где C_{x-1}^{k-1} – количество способов выбрать $k-1$ число из чисел меньше x,

C_{M-1-x}^{n-k} – количество способов выбрать $n-k$ числа из чисел больше x,

C_{M-1}^{n} – общее количество способов выбрать n чисел из множества $\{1,2, \dots, M-1\}$

Тогда математическое ожидание k-й порядковой статистики выражается как:

$$E\left[b_{(k)}\right] = \sum_{x=1}^{M-1} x \frac{C_{x-1}^{k-1} C_{M-1-x}^{n-k}}{C_{M-1}^{n}} = \sum_{x=0}^{M-1} x \frac{C_{x-1}^{k-1} C_{M-1-x}^{n-k}}{C_{M-1}^{n}}$$

Поскольку $C_x^k = \frac{x}{k} C_{x-1}^{k-1}$, получим:

$$E[b_{(k)}] = \frac{k}{C_{M-1}^{n}} \sum_{x=0}^{M-1} C_x^k C_{M-1-x}^{n-k}$$

Воспользуемся известным тождеством $C_n^m = (-1)^{n-m} C_{-(m+1)}^{n-m}$:

$$C_x^k = (-1)^{x-k} C_{-(k+1)}^{x-k}$$

$$C_{M-1-x}^{n-k} = (-1)^{M-1-x-n+k} C_{-(n-k+1)}^{M-1-x-n+k}$$

Подставим в выражение для $E[b_{(k)}]$ и получим

$$E[b_{(k)}] = \frac{k}{C_{M-1}^n} \sum_{x=0}^{M-1} (-1)^{M-1-n} C_{-(k+1)}^{x-k} C_{-(n-k+1)}^{M-1-x-n+k}$$

Теперь воспользуемся сверткой Вандермонда $\sum_i C_r^{q+i} C_s^{p-i} = C_{r+s}^{q+p}$, положив $i = x, r = -(k+1), q = -k, s = -(n-k+1), p = (M-1-n+k)$, тогда

$$E\left[b_{(k)}\right] = \frac{k}{C_{M-1}^n} (-1)^{M-1-n} C_{-n+2}^{M-1-n} = \frac{k}{C_{M-1}^n} C_M^{n+1} = \frac{kM}{n+1}.$$

Из данного утверждения следует, что сильное модульное умножение приводит к значительному увеличению наименьших компонентов формы, особенно при больших значениях модуля *M*. В частности, наименьший компонент $b_{(1)}$ оказывается существенно больше единицы, что делает невозможным покрытие всего диапазона от 1 до $\sum_{i=1}^n b_i$ без пропусков. Хотя случайно может оказаться, что компоненты b_i все же образуют сюръективную или близкую к сюръективной последовательность, вероятность этого пренебрежимо мала.

Таким образом, линейная форма после модульного умножения не будет сюръективной, и, в общем случае, поиск решения для такой формы становится экспоненциально сложной задачей. Из формулы среднего числа решений для всех линейных форм заданной размерности (22), можно получить оценку среднего числа решений уравнения $\sum_{i=1}^n a_i x_i = b$ по всем значениям *b* для формы произвольного вида, которая для больших *M* оказывается существенно меньше 1, что подчеркивает сложность решения этой задачи.

Модульное умножение создает условия для появления разрывов в области значений линейной формы, что вызывает интерес к дальнейшему исследованию этого класса форм. Рассмотрение форм с разрывами позволит лучше понять

влияние модульных преобразований на структуру формы и предложить методы дополнения таких форм для восстановления сюръективности.

2.3 Анализ форм с разрывами в области значений

Исследование линейных форм с разрывами в области значений представляет значительный теоретический интерес в контексте анализа вычислительной сложности. Наличие разрывов свидетельствует о неоднородности распределения решений, что непосредственно влияет на структурную сложность задачи. Изучение зависимости количества разрывов от параметров линейной формы позволяет устанавливать количественные связи между комбинаторными характеристиками задачи и сложностью ее решения. Разработка методов конструктивного построения форм с заданным числом разрывов открывает возможности для целенаправленного формирования классов задач с контролируемой вычислительной сложностью, что имеет важное значение для теоретического анализа границ между легко и трудно разрешимыми случаями задачи о рюкзаке.

Определение 2.4. Разрывом области значений линейной формы $L(x)$ назовем интервал $(d, e) = \{z \in \mathbb{Z}, d < z < e\}$, такой, что $d, e \in L^*(x)$, $\forall z \in (d, e)\ z \notin L^*(x)$.

Определение 2.5. Обозначим через $\nu(L(x_1, \dots, x_n))$ количество разрывов формы $L(x_1, \dots, x_n)$, а через $\mu(L(x_1, \dots, x_n)) = \nu\big(L(x_1, \dots, x_n)\big) + 1$ количество ее отрезков сюръективности.

С точки зрения теоретического анализа сложности, особый интерес представляют линейные формы с разрывами, обладающие регулярной структурой области допустимых значений. Изучение таких структур позволяет исследовать связь между комбинаторными свойствами задачи и вычислительной сложностью ее решения, а также выявлять классы экземпляров с различной алгоритмической поведением.

Определение 2.6. Пусть $[b,c] = \{z \in \mathbb{Z},\ b \le z \le c\} \subseteq L^*(x)$ является подмножеством области значений линейной формы $L(x)$, содержащим все целые числа в отрезке $[b,c]$. Назовем это подмножество отрезком сюръективности линейной формы $L(x)$.

Если не указано обратное, везде далее будем считать, что $a_1 \le a_2 \le \cdots \le a_n$ и $x \in \{0,1\}$.

Определение 2.7. Обозначим через $\nu(L(x_1,\dots,x_n))$ количество разрывов формы $L(x_1,\dots,x_n)$, а через $\mu(L(x_1,\dots,x_n))$ количество ее отрезков сюръективности. Очевидно, что $\mu\big(L(x_1,\dots,x_n)\big) = \nu\big(L(x_1,\dots,x_n)\big) + 1$. Если $L(x_1,\dots,x_n)$ сюръективна, то $\nu\big(L(x_1,\dots,x_n)\big) = 0$, $\mu\big(L(x_1,\dots,x_n)\big) = 1$.

Определение 2.8. Пусть M_1 и M_2 – два множества из целых чисел. Их суммой по Минковскому называется множество $M = M_1 \oplus M_2 = \{a+b, a \in M_1, b \in M_2\}$.

Лемма 2.3. Пусть $L_1\big(x_1,\dots,x_p\big) = \sum_{i=1}^{p} a_i x_i$, $L_2\big(x_{p+1},\dots,x_q\big) = \sum_{i=p+1}^{q} a_i x_i$ и $L_3\big(x_1,\dots,x_q\big) = \sum_{i=1}^{q} a_i x_i$. Тогда $L_3^*\big(x_1,\dots,x_q\big) = L_1^*\big(x_1,\dots,x_p\big) \oplus L_2^*\big(x_{p+1},\dots,x_q\big)$.

Доказательство. По определению $L_3^*\big(x_1,\dots,x_q\big) = \big(\sum_{i=1}^{q} a_i x_i\big)^* = \big(\sum_{i=1}^{p} a_i x_i + \sum_{i=p+1}^{q} a_i x_i\big)^* = \big\{a+b, a \in L_1^*(x_1,\dots,x_p), b \in L_2^*(x_{p+1},\dots,x_q)\big\} = L_1^*(x_1,\dots,x_p) \oplus L_2^*(x_{p+1},\dots,x_q)$.

Таким образом, область определения формы от большего числа переменных можно представить как объединение областей определения первой формы от меньшего числа переменных, смещенных на различные значения из областей определения второй формы $L_3^*\big(x_1,\dots,x_q\big) = \bigcup_{b \in L_2^*(x_{p+1},\dots,x_q)} L_1^*\big(x_1,\dots,x_p\big) \oplus b$. Подобный подход к рассмотрению линейных форм полезен тем, что помогает выявить симметрию относительно значений компонентов формы.

Лемма 2.4. Пусть $L(x_1,\dots,x_{n-1}) = \sum_{i=1}^{n-1} a_i x_i$ и $\mu\big(L(x_1,\dots,x_{n-1})\big) = h$. Если $L(x_1,\dots,x_n) = \sum_{i=1}^{n-1} a_i\, x_i + a_n x_n$, то $\mu\big(L(x_1,\dots,x_n)\big) \le 2\mu(L(x_1,\dots,x_{n-1}))$.

Доказательство. Из леммы 2.3 следует, что $L^*(x_1,\dots,x_n) = L^*(x_1,\dots,x_{n-1}) \oplus (a_n x_n)^* = \{\sum_{i=1}^{n-1} a_i x_i + 0\cdot a_n\} \cup \{\sum_{i=1}^{n-1} a_i x_i + 1\cdot a_n\}$. По условию $\mu\big(\{\sum_{i=1}^{n-1} a_i x_i + 0\cdot a_n\}\big) = \mu\big(L(x_1,\dots,x_{n-1})\big) = h$. Множество значений формы $a_n + \sum_{i=1}^{n-1} a_i x_i$ представляет собой множество значений формы $L(x_1,\dots,x_{n-1})$, смещенное на a_n. Очевидно, что количество отрезков сюръективности при таком смещении сохраняется, поскольку каждый отрезок $[a,b]$ однозначно отображается в отрезок $[a_n + a, a_n + b]$. Следовательно, $\mu\big(\{\sum_{i=1}^{n-1} a_i x_i + 1\cdot a_n\}\big) = \mu(L(x_1,\dots,x_{n-1}))$. Операция объединения множеств может лишь уменьшить число отрезков сюръективности, если эти два множества будут иметь пересекающиеся отрезки, которые объединятся в один. Поэтому $\mu\big(L(x_1,\dots,x_n)\big) = \mu\big(\{\sum_{i=1}^{n-1} a_i x_i + 0\cdot a_n\} \cup \{\sum_{i=1}^{n-1} a_i x_i + 1\cdot a_n\}\big) \le$ $\mu\big(\{\sum_{i=1}^{n-1} a_i x_i + 0\cdot a_n\}\big) + \mu\big(\{\sum_{i=1}^{n-1} a_i x_i + 1\cdot a_n\}\big) = 2\mu(L(x_1,\dots,x_{n-1}))$.

Из данной леммы следует, что при добавлении компонента к форме, ее область значений симметрична на концах, а в середине зависит от пересечения правого конца области $[a_{n-1}, \sum_{i=1}^{n-1} a_i]$ и левого конца области $[a_n, a_n + \sum_{i=1}^{n-1} a_i]$. Таким образом, при добавлении компонента линейной формы, количество ее отрезков сюръективности увеличивается не более чем вдвое.

Лемма 2.5. Пусть $L(x_1,\dots,x_{n-1}) = \sum_{i=1}^{n-1} a_i x_i$ и $\mu\big(L(x_1,\dots,x_{n-1})\big) = h$. Если $L(x_1,\dots,x_n) = \sum_{i=1}^{n-1} a_i\, x_i + a_n x_n$, где $a_n > \sum_{i=1}^{n-1} a_i + 1$, то $\mu\big(L(x_1,\dots,x_n)\big) = 2h$.

Доказательство. В лемме 2 было доказано, что область значений формы $L(x_1,\dots,x_n)$ состоит из объединения множеств значений форм $\sum_{i=1}^{n-1} a_i x_i$ и $a_n + \sum_{i=1}^{n-1} a_i x_i$. Поскольку $a_n > \sum_{i=1}^{n-1} a_i + 1$, отрезки сюръективности форм $\sum_{i=1}^{n-1} a_i x_i$ и $a_n + \sum_{i=1}^{n-1} a_i x_i$ не пересекаются. Поэтому $\mu\big(L(x_1,\dots,x_n)\big) = \mu\big(\{\sum_{i=1}^{n-1} a_i x_i\}\big) + \mu\big(\{\sum_{i=1}^{n-1} a_i x_i + a_n\}\big) = 2\mu(L(x_1,\dots,x_{n-1}))$.

Иными словами, нарушение условия сюръективности (29) на компоненте a_n приведет к раздвоению области значений формы $L(x_1,\dots,x_n)$ на два симметричных подмножества. Таким образом, размер и количество разрывов

области значений линейной формы симметричны относительно середины данной области.

Утверждение 2.6. Пусть $L(x_1, \dots, x_n)$ – линейная форма, для всех компонент которой, кроме k-го выполняется условие сюръективности (29), то есть $a_j \le \sum_{i=1}^{j-1} a_i + 1 \; \forall \, j = \{1, \dots \mathrm{k}-1, \mathrm{k}+1, \dots, n\}$ и $a_k > \sum_{i=1}^{k-1} a_i + 1$. Тогда $\mu\big(L(x_1, \dots, x_n)\big) \le 2^{n-k+1}$.

Доказательство. Рассмотрим последовательность $L(x_1, \dots, x_l) = \sum_{i=1}^{l} a_i x_i \,, l = k-1, \dots, n$ линейных форм, построенных поочередным добавлением компонент формы $L(x_1, \dots, x_n)$. По условию $L(x_1, \dots, x_{k-1})$ сюръективна и из леммы 3 вытекает, что $\mu\big(L(x_1, \dots, x_k)\big) = 2$. Согласно лемме 2, $\mu\big(L(x_1, \dots, x_{k+1})\big) \le \mu\big(L(x_1, \dots, x_k)\big)$, …, $\mu\big(L(x_1, \dots, x_n)\big) \le 2\mu\big(L(x_1, \dots, x_{n-1})\big)$. Подставляя значения из данных неравенств, получим искомое выражение.

Таким образом можно выявить правило для построения линейной формы с заданной сложностью решения. Величина индекса компоненты k, на которой нарушается условие сюръективности, определяет сложность решения формы. Чем раньше происходит нарушение, тем больше число разрывов области значений, и, следовательно, тем сложнее решение.

Лемма 2.6. Пусть $L(x_1, \dots, x_{n-1}) = \sum_{i=1}^{n-1} a_i x_i$, $\mu\big(L(x_1, \dots, x_{n-1})\big) = h$ и $a_t = \min\{a_k : a_k > \sum_{i=1}^{k-1} a_i + 1\}$. Если $L(x_1, \dots, x_n) = \sum_{i=1}^{n-1} a_i \, x_i + a_n x_n$, $a_n \in [\sum_{i=1}^{n-1} a_i - r, \sum_{i=1}^{n-1} a_i + 1]$, где $r = min(a_t - 2, 2\sum_{i=0}^{t-1} a_i)$, то $\mu\big(L(x_1, \dots, x_n)\big) = 2h - 1$.

Доказательство. Область значений $L(x_1, \dots, x_{n-1})$ представляет собой объединение множеств $L^*(x_1, \dots, x_{n-1}) = [0, \sum_{i=1}^{t-1} a_i] \cup [a_t, a_t + \sum_{i=1}^{t-1} a_i] \cup M \cup [\sum_{i=t+1}^{n-1} a_i \,, \sum_{i=t+1}^{n-1} a_i + \sum_{i=1}^{t-1} a_i \,] \cup [a_t + \sum_{i=t+1}^{n-1} a_i \,, \sum_{i=1}^{n-1} a_i]$, где M обозначает множество оставшихся значений формы в промежутке $[a_t + \sum_{i=1}^{t-1} a_i \,, \sum_{i=t+1}^{n-1} a_i]$, то есть $M = L^*(x_1, \dots, x_n) \cup [a_t + \sum_{i=1}^{t-1} a_i \,, \sum_{i=t+1}^{n-1} a_i]$.

Поскольку $a_t > \sum_{i=1}^{t-1} a_i + 1$, интервалы $\big(\sum_{i=1}^{t-1} a_i \,, a_t\big)$ и $\big(\sum_{i=t+1}^{n-1} a_i + \sum_{i=1}^{t-1} a_i \,, \sum_{i=t}^{n-1} a_i\big)$ представляют собой разрывы в области значений формы

$L(x_1, \dots, x_{n-1})$, следовательно, оставшиеся *h – 3* разрыва области значений формы $L(x_1, \dots, x_{n-1})$ приходятся на множество *M.*

Множество значений $L(x_1, \dots, x_{n-1}) + a_n$ представляет собой объединение множеств $(L(x_1, \dots, x_{n-1}) + a_n)^* = [a_n, a_n + \sum_{i=1}^{t-1} a_i] \cup [a_n + a_t, a_n + a_t + \sum_{i=1}^{t-1} a_i] \cup M' \cup [a_n + \sum_{i=t+1}^{n-1} a_i, a_n + \sum_{i=t+1}^{n-1} a_i + \sum_{i=1}^{t-1} a_i] \cup [\sum_{i=t}^{n} a_i, \sum_{i=1}^{n} a_i]$

Множество $M' = M \oplus \{a_n\}$ соответствует множеству *M,* смещенному на a_n и содержит множество значений формы $L(x_1, \dots, x_{n-1}) + a_n$, лежащих в промежутке $[a_n + a_t + \sum_{i=1}^{t-1} a_i, a_n + \sum_{i=t+1}^{n-1} a_i]$. Иначе $M' = (L(x_1, \dots, x_{n-1}) + a_n)^* \cup [a_n + a_t + \sum_{i=1}^{t-1} a_i, a_n + \sum_{i=t+1}^{n-1} a_i]$.

Объединяя эти множества, получаем $L^*(x_1, \dots, x_n) = [0, \sum_{i=1}^{t-1} a_i] \cup [a_t, a_t + \sum_{i=1}^{t-1} a_i] \cup M \cup [\sum_{i=t+1}^{n-1} a_i, \sum_{i=t+1}^{n-1} a_i + \sum_{i=1}^{t-1} a_i] \cup [a_t + \sum_{i=t+1}^{n-1} a_i, \sum_{i=1}^{n-1} a_i] \cup [a_n, a_n + \sum_{i=1}^{t-1} a_i] \cup [a_n + a_t, a_n + a_t + \sum_{i=1}^{t-1} a_i] \cup M' \cup [a_n + \sum_{i=t+1}^{n-1} a_i, a_n + \sum_{i=t+1}^{n-1} a_i + \sum_{i=1}^{t-1} a_i] \cup [\sum_{i=t}^{n} a_i, \sum_{i=1}^{n} a_i]$.

Рассмотрим сначала случай $a_t - 2 \leq 2\sum_{i=1}^{t-1} a_i$. По условию $a_n \geq \sum_{i=1}^{n-1} a_i - a_t + 2$, следовательно отрезок $[\sum_{i=t+1}^{n-1} a_i, \sum_{i=t+1}^{n-1} a_i + \sum_{i=1}^{t-1} a_i]$ лежит левее отрезка $[a_n, a_n + \sum_{i=1}^{t-1} a_i]$, поэтому области $L^*(x_1, \dots, x_n) \cup [0, \sum_{i=t+1}^{n-1} a_i + \sum_{i=1}^{t-1} a_i]$ и $(L(x_1, \dots, x_{n-1}) + a_n)^*$ не пересекаются.

При этом $a_n + \sum_{i=1}^{t-1} a_i \geq \sum_{i=t}^{n-1} a_i + 2\sum_{i=1}^{t-1} a_i - a_t + 2 \geq \sum_{i=t+1}^{n-1} a_i$, поэтому отрезки $[a_t + \sum_{i=t+1}^{n-1} a_i, \sum_{i=1}^{n-1} a_i]$ и $[a_n, a_n + \sum_{i=1}^{t-1} a_i]$ пересекаются.

Вместе с тем $a_n + a_t \geq \sum_{i=1}^{n-1} a_i + 2$, так что отрезок $[a_t + \sum_{i=t+1}^{n-1} a_i, \sum_{i=1}^{n-1} a_i]$ расположен левее отрезка $[a_n + a_t, a_n + a_t + \sum_{i=1}^{t-1} a_i]$, следовательно области $L^*(x_1, \dots, x_n)$ и $(L(x_1, \dots, x_{n-1}) + a_n)^* \cup [a_n + a_t, \sum_{i=1}^{n} a_i]$ не пересекаются.

Подводя итог, получаем, что области значений $L^*(x_1, \dots, x_n)$ и $(L(x_1, \dots, x_{n-1}) + a_n)^*$ пересекаются только по одному интервалу, поэтому $\mu\big(L^*(x_1, \dots, x_n)\big) = \mu\big(L^*(x_1, \dots, x_n)\big) + \mu(L(x_1, \dots, x_{n-1}) + a_n)^* - 1 = 2h - 1$.

Теперь рассмотрим случай $a_t - 2 > 2\sum_{i=1}^{t-1} a_i$. По условию $a_n \geq \sum_{i=1}^{n-1} a_i - 2\sum_{i=1}^{t-1} a_i > \sum_{i=1}^{n-1} a_i - a_t + 2$, следовательно, отрезок

$[\sum_{i=t+1}^{n-1} a_i\,, \sum_{i=t+1}^{n-1} a_i + \sum_{i=1}^{t-1} a_i\,]$ лежит левее отрезка $[a_n, a_n + \sum_{i=1}^{t-1} a_i]$, поэтому области $L^*(x_1, \dots, x_n) \cup [0, \sum_{i=t+1}^{n-1} a_i + \sum_{i=1}^{t-1} a_i\,]$ и $(L(x_1, \dots, x_{n-1}) + a_n)^*$ не пересекаются.

При этом $a_n + \sum_{i=1}^{t-1} a_i \geq \sum_{i=1}^{n-1} a_i - \sum_{i=1}^{t-1} a_i = \sum_{i=\mathrm{t}}^{n-1} a_i > \sum_{i=t+1}^{n-1} a_i$, поэтому отрезки $[a_t + \sum_{i=t+1}^{n-1} a_i\,, \sum_{i=1}^{n-1} a_i]$ и $[a_n, a_n + \sum_{i=1}^{t-1} a_i]$ пересекаются.

Вместе с тем $a_n + a_t \geq \sum_{i=1}^{n-1} a_i - 2\sum_{i=1}^{t-1} a_i + a_t \geq \sum_{i=1}^{n-1} a_i + 2$, так что отрезок $[a_t + \sum_{i=t+1}^{n-1} a_i\,, \sum_{i=1}^{n-1} a_i]$ расположен левее отрезка $[a_n + a_t, a_n + a_t + \sum_{i=1}^{t-1} a_i]$, следовательно области $L^*(x_1, \dots, x_n)$ и $(L(x_1, \dots, x_{n-1}) + a_n)^* \cup [a_n + a_t, \sum_{i=1}^{n} a_i]$ не пересекаются.

Подводя итог, получаем, что области значений $L^*(x_1, \dots, x_n)$ и $(L(x_1, \dots, x_{n-1}) + a_n)^*$ пересекаются только по одному интервалу, поэтому $\mu\big(L^*(x_1, \dots, x_n)\big) = \mu\big(L^*(x_1, \dots, x_n)\big) + \mu(L(x_1, \dots, x_{n-1}) + a_n)^* - 1 = 2h - 1$.

Данная лемма позволяет не только определить число разрывов в области значений формы, но и оценить их ширину.

Обобщая результаты предыдущих лемм, получим процедуру построения линейной формы с заданным числом разрывов в области значений.

Теорема 2.7. Существует полиномиальный алгоритм, позволяющий любого $h \in [0, 2^n - 1]$ построить линейную форму $L(x_1, \dots, x_n)$ с h разрывами.

Доказательство. Докажем теорему, описав процедуру построения такой формы. Разложим h в сумму степеней двойки: $h = \sum_{i=0}^{n-1} 2^i y_i$.

Найдем $k = \max\{i : y_i = 1\}$ и положим $t = n - k$. Если $t > 1$, возьмем $a_1 = 1$ и $\forall\, j = 2, \dots, t-1$ выберем $a_j \in \big[a_j - 1, \sum_{i=1}^{j-1} a_i + 1\big]$. Таким образом мы построили сюръективную линейную форму $L(x_1, \dots, x_t)$.

Далее выбираем $a_t > \sum_{i=1}^{t-1} a_i + 1$ и находим $r = min(a_t - 2, 2\sum_{i=0}^{t} a_i)$. На этом шаге имеем $\nu\big(L(x_1, \dots, x_{t+1})\big) = 1$.

Затем для $\forall\, j = t+1, \dots, n$ если $y_{n-j} = 1$, выбираем $a_j > \sum_{i=1}^{j} a_i + 1$ и в соответствии с леммой 2.5, получаем $\nu\left(L\big(x_1, \dots, x_j\big)\right) = 2\nu\left(L\big(x_1, \dots, x_{j-1}\big)\right) + 1$,

иначе выбираем $a_j \in \left[\sum_{i=1}^{j-1} a_i - r, \sum_{i=1}^{j-1} a_i + 1\right]$ и по лемме 2.6 имеем $\nu\left(L(x_1, \dots, x_j)\right) = 2\nu\left(L(x_1, \dots, x_{j-1})\right)$.

В результате, на каждом шаге j происходит удвоение числа разрывов, а если $y_{n-j} = 1$, то к числу разрывов прибавляется 1, которая к концу процедуры удваивается $n - j$ раз. Таким образом получаем $\nu\big(L(x_1, \dots, x_n)\big) = \sum_{i=0}^{n-1} 2^i y_i = h$

Предложенный в доказательстве теоремы алгоритм построения линейной формы с заданным числом разрывов обладает полиномиальной сложностью и может быть использован для генерации тестовых примеров с контролируемыми характеристиками. Полученные результаты важны не только для решения прикладных задач, но и для фундаментального исследования задачи о рюкзаке в различных постановках.

Для анализа свойств таких форм может быть применен метод дополнения, позволяющий устранить разрывы и восстановить сюръективность. Данный подход основан на итеративном добавлении компонентов, удовлетворяющих условиям сюръективности, в позиции, где эти условия нарушаются. Процедура начинается с нахождения минимального индекса t, для которого нарушается условие сюръективности: $a_t = \min\{a_k : a_k > \sum_{i=1}^{k-1} a_i + 1\}$ с последующим добавлением компонента $v_i = \sum_{i=1}^{t-1} a_i + 1$. Такой метод обеспечивает минимальность числа добавляемых компонентов.

Полученная дополненная форма удовлетворяет условиям сюръективности, что позволяет применять к ней эффективные алгоритмы решения. При этом в алгоритм может быть добавлен шаг фильтрации, отбрасывающий те ветви решений, в которых значения компонент решения, соответствующие добавленным компонентам формы, отличны от нуля. Для форм с ограниченным числом решений такой подход сохраняет высокую эффективность.

Таким образом, задачи с линейными формами, содержащими разрывы, могут быть решены за приемлемое время при соответствующем выборе коэффициентов. Использование таких форм расширяет множество исследуемых комбинаторных структур по сравнению с классом сюръективных форм, что

представляет дополнительный интерес для теоретического анализа сложности задачи о рюкзаке.

Выводы к главе 2

Сюръективные линейные формы представляют теоретический интерес как класс комбинаторных объектов, для которых задача о рюкзаке допускает эффективное решение. Вычислительная сложность нахождения решений для таких форм оценивается как *O(ns)*, где *n* — размерность задачи, а *s* — количество решений, что существенно ниже сложности решения общей NP-трудной постановки задачи.

Важным свойством сюръективных форм является их высокая плотность, превышающая критическое значение 0,9408, что делает их устойчивыми к решению на основе редукции решеток. Поскольку коэффициенты сюръективных векторов ограничены снизу значением *n*, их плотность удовлетворяет неравенству $d(A) \geq \frac{n}{n-1} > 1$, что исключает возможность применения алгоритмов, эффективных для низкоплотностных случаев.

Основной особенностью сюръективных форм является их неинъективность, приводящая к множественности решений. Данное свойство требует разработки специализированных методов обработки, таких как введение избыточности и механизмов проверки корректности решений.

Таким образом, исследование сюръективных линейных форм вносит вклад в понимание границы между полиномиально разрешимыми и NP-трудными случаями задачи о рюкзаке. Дальнейшие исследования могут быть направлены на разработку эффективных алгоритмов работы с такими формами и анализ их свойств в контексте общей теории сложности вычислений.

3 Построение и анализ рюкзачных структур с управляемой сложностью решения

В данной главе исследуется влияние параметров задачи рюкзачного типа: длины и плотности вектора, а также числа допустимых решений — на вычислительную сложность ее решения. При низкой плотности экземпляры эффективно решаются методами редукции решеток, включая алгоритм Ленстры–Ленстры–Ловаса (LLL), однако при увеличении плотности их результативность резко снижается. В то же время исследованные в предыдущей главе классы экземпляров сохраняют полиномиальную разрешимость при высокой плотности векторов, что позволяет выделить в отдельный класс легкорешаемых экземпляров, сохраняющих трудность для решения методом редукции базиса решеток.

Особое внимание уделено задаче конструирования экземпляров с управляемыми характеристиками сложности, позволяющими экспериментально исследовать границу между полиномиально и экспоненциально разрешимыми случаями. Разработанные алгоритмы обеспечивают генерацию экземпляров, для которых методы редукции решеток, включая LLL, теряют эффективность, тогда как предложенный в работе подход сохраняет возможность восстановления решений за полиномиальное время.

Дополнительно анализируется влияние модульных преобразований на структуру и сложность задачи. Показано, что при модульных преобразованиях сохраняется множество допустимых решений, однако сама задача становится труднорешаемой для известных алгоритмов, что позволяет использовать такие преобразования в прикладных задачах защиты информации.

В заключительной части главы представлена практическая реализация разработанных алгоритмов и результаты вычислительных экспериментов, подтверждающие теоретические выводы. Программный комплекс включает модули для построения рюкзачных последовательностей, анализа параметров,

проверки корректности решений и моделирования поведения различных алгоритмов. Показано, что разработанные методы можно применять при конструировании криптографических примитивов и предложена асимметричная схема шифрования на их основе. Полученные в экспериментах результаты подтверждают корректность теоретических положений и демонстрируют возможность управлять вычислительной сложностью рюкзачных задач путем вариации их параметров.

3.1 Управление сложностью решения через структурные параметры рюкзачных векторов

Как показано в предыдущих главах, особое значение в анализе задачи о рюкзаке имеет параметр плотности $d(A) = \frac{n}{\log_2(\max_i a_i)}$, характеризующий соотношение между размерностью задачи и диапазоном значений коэффициентов. При плотности ниже критического порога ($d < 0{,}94$) методы редукции решеток эффективно восстанавливают решения, поскольку пространство допустимых векторов сжато и содержит множество коротких векторов, приближенных к оптимальному. При $d(A) > 1$ задача переходит в область экспоненциальной сложности, где приближенные алгоритмы теряют точность.

В то же время, анализ структуры множества допустимых решений, выполненный во второй главе, показал, что даже для фиксированного $d(A)$ структура множества решений может существенно меняться в зависимости от способа формирования коэффициентов a_i. Например, сюръективные формы характеризуются равномерным покрытием диапазона значений правой части и допускают эффективный поиск решений, тогда как формы с разрывами обладают избыточными или пропущенными значениями, что резко увеличивает вычислительные затраты.

На практике это означает, что одна и та же задача может быть либо тривиальной, либо сложной для решения существующими алгоритмами, в

зависимости от того, как заданы ее параметры. Поэтому третья глава направлена на выявление закономерностей, позволяющих подбирать параметры так, чтобы задача оставалась разрешимой предложенными аналитическими методами, но при этом не сводилась к простому случаю при использовании стандартных приближенных подходов, таких как LLL-редукция.

Такой подход позволяет не только уточнить границы применимости алгоритмов, но и выделить практический класс задач, обладающих устойчивым поведением при применении конкретных алгоритмов. Как будет показано далее, эти результаты могут быть использованы при построении схем кодирования и защиты информации.

Выбор длины вектора n. Одним из основных параметров, определяющих вычислительную сложность задачи рюкзачного типа, является длина вектора коэффициентов n. Изменение этого параметра влияет не только на размерность пространства решений, но и на его структуру и эффективность различных алгоритмов решения задачи.

Известным классом легкорешаемых экземпляров задач рюкзачного типа являются сверхрастущие последовательности, лежащие в основе схемы Меркла–Хеллмана [56]. Эти последовательности образуют особый класс экземпляров, для которых задача имеет простую и детерминированную структуру и потому решается за полиномиальное время. В схеме Меркла–Хеллмана использовались векторы длиной $n = 100$. Более поздние исследования [57, 58] показали, что увеличение размерности до $n = 150$ и выше приводит к существенному изменению характера распределения решений и снижению эффективности методов редукции решеток при фиксированной плотности.

В настоящей работе для анализа свойств экземпляров с высокой плотностью рассматриваются задачи размерности $n = 200$, где элементы вектора после модульного преобразования принимают значения в диапазоне от 2^{170} до 2^{200}. Данное соотношение обеспечивает плотность, близкую к критическому значению $d \approx 1$, что соответствует теоретической области наибольшей

вычислительной сложности задачи. Такой выбор параметров позволяет исследовать широкий спектр характеристик экземпляров для анализа эффективности различных алгоритмов и наблюдать переход от легкорешаемых к труднорешаемым случаям.

Как показано в дальнейшем, рост n приводит к увеличению диапазона коэффициентов и, как следствие, снижению плотности рюкзака при фиксированном среднем числе решений. Это изменение параметров напрямую влияет на вычислительную сложность: при недостаточной плотности задачи становятся чувствительными к редукции решеток, а при избыточной — теряют структурную упорядоченность, что затрудняет поиск решений любыми методами. При плотностях $d < 0{,}94$ задача о рюкзаке эффективно решается методом LLL [44]; при $d > 1$ нарушается однозначность отображения множества решений. Наиболее сложные случаи наблюдаются при $d \approx 1 + \frac{log\frac{n}{2}}{n}$ [59], что подтверждается как теоретическими, так и численными оценками. Для выбранной размерности $n = 200$ данное условие соответствует плотности $d \approx 1{,}033$. Такое значение плотности выбрано в дальнейшем для последующих расчетов.

Теоретическая оценка трудоемкости базовой реализации LLL-алгоритма при $d \approx 1$ выражается как

$$T(n) = O(n^6 \log^3 a_n) \approx O(n^9),$$

где a_n — величина старшего коэффициента вектора. Для размерностей порядка $n \geq 200$ вычислительные затраты становятся настолько велики, что практическое применение метода для полного восстановления структуры задачи оказывается нецелесообразным. Таким образом, увеличение размерности рюкзачного вектора не только расширяет пространство допустимых решений, но и существенно повышает устойчивость задачи к методам редукции базиса, обеспечивая переход от эффективно решаемых к практически неразрешимым экземплярам.

В то же время, для специализированных классов рюкзачных векторов могут существовать более эффективные алгоритмы. В частности, для сюръективных форм получена следующая нижняя оценка сложности для разработанного в главе 2 алгоритма.

Утверждение 3.1. Сложность решения задачи о рюкзаке в форме распознавания $\sum_{i=1}^{n} a_i x_i = b$ с сюръективной формой A при заданной плотности d в среднем случае ограничена снизу оценкой $\Omega\left(n * 2^{\left(n\left(1-\frac{1}{d}\right)-1\right)}\right)$.

Доказательство. Поскольку форма $L(x_1, \dots, x_n) = \sum_{i=1}^{n} a_i x_i$ сюръективна и принимает все значения в интервале $[0, \sum_{i=1}^{n} a_i]$, ее среднее число решений по всем b в данном интервале выражается как $N(L) = \frac{2^n}{\sum_{i=1}^{n} a_i + 1}$. При заданной плотности d из формулы $d(A) = \frac{n}{\log_2 \max A} = \frac{n}{\log_2 a_n}$ можно выразить $a_n = 2^{\frac{n}{d}}$. Подставляя это в выражение для среднего числа решений и учитывая условия сюръективности, получаем $N(l) = \frac{2^n}{\sum_{i=1}^{n} a_i + 1} \geq \frac{2^n}{2a_n} = 2^{n\left(1-\frac{1}{d}\right)-1}$.

Таким образом, с ростом плотности рюкзачного вектора сложность решения задачи со сюръективной формой существенно возрастает. При значениях плотности $d \gg 1$ нижняя оценка сложности $\Omega\left(n * 2^{\left(n\left(1-\frac{1}{d}\right)-1\right)}\right)$ приближается к $\Omega\left(2^{(n-1)}\right)$, что свидетельствует об экспоненциальной сложности решения даже для этого структурно простого класса задач. Данный результат демонстрирует принципиальные ограничения эффективности алгоритмов решения задачи о рюкзаке в области высокой плотности и подтверждает необходимость тщательного подбора параметров при исследовании вычислительной сложности.

Поскольку в рамках рассматриваемого подхода среднее число решений выбирается минимальным из возможных при фиксированном значении плотности d, можно считать, что асимптотическая сложность в среднем случае приближается к нижней оценке, выражаемой как

$$T_{\text{avg}}(n,d) = \Theta(n \cdot 2^{\,n(1-1/d)-1}).$$

Как будет показано далее, хотя сложность решения сюръективных форм является линейной относительно размерности вектора и числа решений, для сохранения плотности $d \approx 1{,}033$ при $n > 200$ необходимо выбрать значительно большее число средних решений, из-за чего требуемые вычислительные ресурсы существенно возрастают. Таким образом, выбор размерности $n = 200$ представляет собой разумный компромисс между обеспечением устойчивости к редукции решеток и практической разрешимостью.

Определение плотности рюкзачного вектора. На основе эмпирического теста в работе [60], установлено, что задачи о рюкзаке наибольшей сложности возникают при плотности рюкзачного вектора около $1 + \frac{\log_2 \frac{n}{2}}{n}$, что часто приводит к наличию нескольких решений, особенно при правой части уравнения, близкой к $\frac{1}{2}\sum_{i=1}^{n} a_i$.

В работе [59] отмечается, что для экземпляров размерности $n \approx 100$ и плотности $d \approx 1$ не известно эффективных алгоритмов решения, особенно при весе решения $\approx \frac{n}{2}$. Это позволяет предположить, что экземпляры с плотностью в интервале между 1 и значением, указанным в [60], характеризуются повышенной вычислительной стойкостью. При выбранном ранее значении $n = 200$ формула $d = 1 + \frac{\log_2(n/2)}{n}$ дает $d \approx 1{,}033$, что обеспечивает существенное усложнение решения задачи и повышает устойчивость к алгоритмам, основанным на редукции решеток.

В базовой конструкции начальные элементы сюръективной последовательности задаются детерминировано: $a_1 = 1, a_2 = 2$, а для малых k рекуррентная граница для следующего коэффициента вычисляется как $a_k \in [a_{k-1} + 1;\ \sum_{i=1}^{k-1} a_i + 1]$. При таких значениях первые члены сюръективной последовательности обладают двумя важными свойствами: во-первых, разные варианты этих членов незначительно отличаются между собой, поэтому число различных комбинаций этих членов невелико, и они малоинтересны с точки

зрения анализа сложных случаев. Во-вторых, при использовании подобных предсказуемых начальных фрагментов в прикладных схемах (например, для защиты информации) они облегчают восстановление исходной последовательности: анализ этих тривиальных коэффициентов предоставляет атакующему опорную информацию для восстановления дальнейших параметров. По этим причинам в процедуре генерации фиксируется короткий детерминированный начальный фрагмент длины p, а последующая часть формы строится по предложенному алгоритму, начиная с p-го элемента. В частности, для текущей реализации $p = 5$.

Плотность при этом будет оцениваться по эффективной длине $n' = n - p$:

$$d(A) \approx \frac{n - p}{\log_2(\max A)},$$

что корректно отражает вклад именно случайной части последовательности в формирование вычислительной сложности экземпляра.

Выбор значения модуля M при модульном преобразовании оказывает существенное влияние на комбинаторные свойства задачи о рюкзаке и, в частности, на ее вычислительную сложность. Сильное модульное умножение, нарушая сюръективность исходной линейной формы, приводит к тому, что часть значений правой части уравнения становится недостижимой. Тем не менее, структура множества допустимых значений сохраняется, что делает данный тип преобразований особенно интересным для анализа перехода от легкорешаемых к труднорешаемым экземплярам задачи [61].

Как было показано в утверждении 2.5, при применении модульного умножения плотность преобразованного вектора можно оценить выражением: $d(A) \approx \frac{n'}{\log_2 M}$. При этом значение M выступает параметром, определяющим баланс между плотностью и сложностью решения. Слишком большое значение модуля Mприводит к уменьшению плотности $d(A)$, что влечет за собой упрощение структуры задачи и повышению вероятности ее успешного решения методами редукции базиса решетки. В то же время слишком малые значения M

нарушают исходную структуру множества решений и приводят к появлению вырожденных экземпляров.

Из этого следует, что для достижения желаемой плотности $d^* \geq 1{,}033$ при длине вектора $n = 200$ и $p = 5$ модуль M должен удовлетворять условию: $M \leq 2^{\frac{n'}{d}} = 2^{\frac{195}{1{,}033}} \approx 2^{189}$. При этом для сохранения мнодества решений требуется выбирать $M \geq \sum_{i=1}^{n} a_i$. В частности, в работе [62] предложено использовать $M \approx \sum_{i=1}^{n} a_i$. В текущей реализации было принято решение выбирать $M \approx \sum_{i=1}^{n} a_i + k$, где $k \in [0;\ a_{n-1}]$, поскольку для больших n такая надбавка k на порядок меньше значения $\sum_{i=1}^{n} a_i$ и не оказывает существенного влияния на величину плотности. Такой подход сохраняет исходное множество решений, обеспечивая при этом высокую плотность преобразованного экземпляра.

Выбор среднего числа решений. Основной недостаток алгоритма решения, приведенного в теореме 2, заключается в том, что число допустимых решений уравнения $\sum_{i=1}^{n} a_i x_i = b$ с сюръективной формой может значительно превышать величину n. Например, для формы $L(x_1, \dots, x_{100}) = \sum_{i=1}^{100} i\, x_i$ число решений задачи $L(x_1, \dots, x_{100}) = 2525$ приблизительно равно 1,73e+27, что во много раз превышает значение n.

Однако при построении сюръективной формы можно ограничить среднее число ее решений. Поскольку среднее число решений сюръективной формы $L(x_1, \dots, x_n)$ по всем b в интервале $[0;\ \sum_{i=1}^{n} a_i]$ выражается как $N(L) = \frac{2^n}{\sum_{i=1}^{n} a_i + 1}$, чтобы среднее число решений не превышало заданного значения $\widetilde{N}$ и распределение числа решений по интервалу $[0;\ \sum_{i=1}^{n} a_i]$ было близко к равномерному, достаточно при ее построении выбирать каждый следующий a_k так, чтобы выполнялось $a_k \geq \frac{2^{k-1}}{\widetilde{N}}$.

Таким образом, можно выбирать коэффициенты линейной формы так, чтобы сложность алгоритма нахождения ее решений не превышала выбранного значения $O(\widetilde{N}n)$. Плотность рюкзака, построенного по описанной процедуре,

будет $d(A) \leq \frac{n}{(n-1)-\log_2 \widetilde{N}}$. Это позволяет устанавливать контролируемое соотношение между плотностью вектора и средним числом решений, что обеспечивает возможность конструктивного формирования экземпляров задач с заданными характеристиками вычислительной сложности.

Для обеспечения желаемой плотности $d^* \approx 1{,}033$ необходимо учитывать следующие соотношения: $2^{189} \geq M \geq \sum_{i=1}^{n} a_i \geq 2a_n \geq \frac{2^n}{\widetilde{N}}$. Из этого выражения можно вывести, что $\widetilde{N} \geq \frac{2^{200}}{2^{189}} = 2048$.

С учетом выполнения условий сюръективности, плотность рюкзака, построенного по такой процедуре, будет ограничена неравенствами $\frac{n}{(n-1)} \leq d(A) \leq \frac{n}{(n-1)-\log_2 \widetilde{N}}$. Полученная плотность рюкзачного вектора превышает критическое значение 0,94, что делает невозможным эффективное применение алгоритмов, разработанных для низкоплотностных случаев, таких как метод Костера-Лагариаса-Одлыжко. Поскольку современные алгоритмы решения задачи о рюкзаке, включая методы редукции решеток, демонстрируют высокую эффективность преимущественно при плотностях ниже единицы, использование сконструированных векторов с плотностью $d > 1$ создает принципиальные вычислительные барьеры. Это позволяет формировать классы экземпляров задачи, сохраняющие высокую сложность решения даже при использовании современных оптимизированных алгоритмов.

Выбор контрольной суммы. При исследовании задач о рюкзаке с большим числом допустимых решений возникает проблема неоднозначности — разные комбинации переменных могут давать одно и то же значение правой части уравнения. Для устранения этой неоднозначности вводится небольшая избыточность в виде контрольной последовательности, позволяющей однозначно идентифицировать корректное решение среди множества допустимых.

Так как старшим битам вектора переменных соответствуют более крупные коэффициенты формы, совпадение разных решений по старшим позициям

встречается значительно чаще, чем по младшим. Поэтому контрольную последовательность целесообразно размещать в начале решения, чтобы обеспечить чувствительность метода к минимальным отклонениям в младших разрядах. Вероятность случайного совпадения контрольной последовательности в разных решениях можно оценить выражением 2^{-l}, где l— ее длина в битах.

Для уменьшения вычислительных затрат и сохранения компактности формата решения контрольная последовательность формируется с помощью усеченного хэш-преобразования из первых l бит стандартной функции SHA-256 [63]. При $l = 16$ с учетом отброшенных ранее $p = 5$ элементов вероятность случайного совпадения контрольных последовательностей в различных решениях составляет порядка $2^{-21} = 1/2097152$, что обеспечивает высокую надежность идентификации.

В задачах с параметрами $n = 200$, $l = 16$, $p = 5$ избыточная часть данных (включая контрольную последовательность и пропущенные элементы) составляет лишь около 10,5% от общего объема. Это позволяет сохранить компактность представления решений и одновременно обеспечивает корректность восстановления при множественности допустимых комбинаций.

3.2 Применение полученных результатов в прикладных задачах защиты информации

Разработанные в предыдущих разделах методы анализа комбинаторных структур и построения линейных форм с управляемыми свойствами открывают возможности их применения в различных прикладных задачах. Одной из наиболее показательных областей использования задач рюкзачного типа является теория кодирования и криптографические протоколы, в которых вычислительная сложность играет ключевую роль в обеспечении стойкости и корректности процедур преобразования данных [64, 65].

Задача о рюкзаке в форме распознавания послужила основой для построения ряда схем, использующих принцип перехода от легко решаемых экземпляров к труднорешаемым путем параметрических преобразований. Такие

конструкции демонстрируют, как комбинаторные свойства задачи могут быть использованы для управления сложностью и устойчивостью алгоритмов.

В настоящем разделе рассматриваются возможности применения разработанных структурных подходов на примере криптосистемы Меркла–Хеллмана, в которой сюръективные формы выступают как перспективная основа для построения преобразований, обеспечивающих управляемую сложность и однозначность восстановления решений.

В криптографических приложениях рассматривается следующая вариация задачи о рюкзаке: для заданного множества чисел $S = \{a_1, a_2, \dots, a_n\}$ и целевого значения b требуется найти вектор $x = (x_1, x_2, \dots, x_n)$, где $x_i \in \{0, 1\}$, где $x_i \in \{0, 1\}$, такой, что выполняется равенство $\sum_{i=1}^{n} a_i x_i = b$.

В рюкзачных криптосистемах вектор *A* используется для зашифрования блока открытого текста *x*, состоящего из *n* двоичных символов. Поскольку желательна однозначность расшифрования, для зашифрования информации, как правило, используют инъективные рюкзачные векторы.

Первой асимметричной рюкзачной криптосистемой считается криптосистема Меркла-Хеллмана [56]. В этой системе секретным ключом является сверхрастущий рюкзачный вектор, к которому затем применяется запутывающее преобразование.

Из определения, приведенного ранее, следует, что любой сверхрастущий вектор является также инъективным. Использование сверхрастущих векторов обеспечивает простоту расшифрования, поскольку для сверхрастущего набора чисел решение задачи может быть найдено за линейное время последовательным извлечением значений x_i.

В криптосистеме Меркла-Хеллмана сверхрастущий вектор $A = (a_1, \dots, a_n)$ преобразуется в инъективный вектор $B = (b_1, \dots, b_n)$ с помощью сильного модульного умножения на простой множитель *u* относительно взаимно простого с *u* модуля *M*, $1 < u < M$. Обратное преобразование выполняется с помощью сильного модульного умножения на число $t = u^{-1} mod\, M$. Набор

(A, u, t, M) является закрытым ключом криптосистемы, а вектор B – открытым ключом. Сообщение представляется вектором $x = (x_1, x_2, \dots, x_n)$, где $x_i \in \{0, 1\}$. Шифрование выполняется путем вычисления суммы $C = \sum_{i=1}^{n} b_i x_i$. Получатель, обладая закрытым ключом, использует значение t для расшифрования: $C' = (C * t) mod\ M$, затем решает задачу $\sum_{i=1}^{n} a_i x_i = C'$ итеративно устанавливая $x_i = 1$, если $C' - \sum_{j=i+1}^{n} a_j x_j \geq a_i$ и $x_i = 0$ в противном случае для $i = n, \dots, 1$.

Алгоритм 1. Генерация ключей в схеме Меркла-Хеллмана

Входные параметры:

- длина последовательности n;
- начальный диапазон генерации $[a'; a'']$;

Вывод:

- закрытый ключ (A, M, u);
- открытый ключ $B = (b_1, b_2, \dots, b_n)$.

Шаг 1. Сгенерировать сверхрастущую последовательность: $A = (a_1, a_2, \dots, a_n)$, $a_1 \in [a'; a'']$, $a_k > \sum_{i=1}^{k-1} a_i \ \forall\ k = 2, \dots, n$.

Шаг 2. Выбрать модуль M, удовлетворяющий условию сильного модульного умножения: $M > \sum_{i=1}^{n} a_i$.

Шаг 3. Выбрать множитель u, взаимно простой с M: $\gcd(u, M) = 1$.

Шаг 4. Вычислить открытый вектор: $b_i = (u a_i) mod M, i = 1, \dots, n$.

Шаг 5. Вернуть пары ключей: открытый ключ $B = (b_1, b_2, \dots, b_n)$, закрытый ключ (A, M, u).

Алгоритм 2. Шифрование сообщения в схеме Меркла-Хеллмана

Входные параметры:

- открытый ключ $B = (b_1, b_2, \dots, b_n)$;
- бинарное сообщение $x = (x_1, x_2, \dots, x_n), x_i \in \{0, 1\}$

Вывод:

- зашифрованное сообщение $C \in \mathbb{Z}_M$.

Шаг 1. Инициализация суммы: $C := 0$.

Шаг 2. Для каждого $i = 1,2, \dots, n$: если $x_i = 1$, то $C := C + b_i$.

Шаг 3. Привести сумму по модулю: $C := C \ mod M$.

Шаг 4. Вернуть значение $C = \sum_{i=1}^{n} b_i x_i \ mod M$.

Вектор x не может быть восстановлен без знания параметров A, u, M, поскольку структура сверхрастущего вектора A скрыта преобразованием.

Алгоритм 3. Расшифрование сообщения в схеме Меркла-Хеллмана

Входные параметры:

- зашифрованное сообщение C;
- закрытый ключ (A, M, u).

Вывод:

- восстановленное сообщение $x = (x_1, x_2, \dots, x_n)$.

Шаг 1. Найти обратный элемент t к u по модулю M: $t = u^{-1} mod M$.

Шаг 2. Вычислить промежуточное значение: $C' = (C \cdot t) mod M$.

Шаг 3. Решить задачу восстановления исходного вектора из уравнения $\sum_{i=1}^{n} a_i x_i = C'$. Для этого выполнить итеративный алгоритм, начиная с $i = n$ и двигаясь к 1:

- если $C' \geq a_i$, то положить $x_i = 1$ и обновить $C' := C' - a_i$;
- иначе положить $x_i = 0$.

Шаг 4. Возвратить восстановленный вектор $x = (x_1, x_2, \dots, x_n)$.

Использование сверхрастущего вектора A обеспечивает возможность восстановления решения x за время $O(n)$, что гарантирует эффективность обратного преобразования при наличии секретных параметров (A, M, u). Однако в условиях, когда известны только публичные параметры (B, C), задача сводится к решению классической NP-трудной задачи о рюкзаке.

Несмотря на потенциал NP-полных задач рюкзачного типа в криптографии, анализ классических рюкзачных систем, включая схему Меркла–Хеллмана, выявил их уязвимость, обусловленную структурной предсказуемостью используемых векторов. Применение сверхрастущих последовательностей, несмотря на обеспечение эффективного обратного преобразования, создает структурную предсказуемость, позволяющую проводить криптоанализ, как в

атаке Шамира [66], эксплуатирующей взаимосвязь между открытым и закрытым ключами.

Еще одним параметром, влияющим на криптостойкость рюкзачных схем, является плотность укладки $d(A) = \frac{n}{\log_2 \max A}$. При низкой плотности $d \leq 0{,}9408$ задача о рюкзаке становится решаемой с помощью методов редукции решеток, в частности алгоритма Ленстры–Ленстры–Ловаса (LLL) [44]. Это делает рюкзаки с низкой плотностью неэффективными для практического использования в условиях современных угроз.

Для преодоления этих ограничений в данной работе предлагается переход к использованию плотных рюкзаков, характеризующихся значением плотности, превышающим единицу. Такие рюкзаки затрудняют применение известных криптоаналитических методов, включая методы редукции решеток, и обладают более сложной структурой множества допустимых решений. В ряде исследований было установлено, что именно при таких параметрах задача о рюкзаке приобретает большую сложность [67]. В частности, в работе [60] показано, что задачи о рюкзаке в форме распознавания становятся особенно сложными при плотности $d \approx 1 + \frac{log\frac{n}{2}}{n}$. Такая плотность, наряду с правой частью уравнения, близкой к $\frac{1}{2}\sum_{i=1}^{n} a_i$, приводит к множественности допустимых решений, существенно затрудняющей дешифрование даже при наличии частичной информации о ключах. Также установлено, что большинство сообщений с весом около $n/2$ в таких задачах имеют несколько альтернативных расшифровок с близким весом, что создает дополнительную защиту от атак, направленных на точную реконструкцию исходного сообщения.

Анализ исследований [59] подтверждает отсутствие успешных атак на задачи распознавания при размере рюкзака $n \approx 100$, плотности $d \approx 1$ и весе сообщения, близком к $n/2$. Это указывает на наличие класса задач, устойчивых к современным криптоаналитическим методам, и подтверждает их потенциальную пригодность для построения защищенных протоколов.

Особое внимание заслуживают неинъективные рюкзаки, в которых разные векторы x могут давать одинаковую сумму $\sum_{i=1}^{n} a_i x_i$. Такие векторы усложняют задачу обратного восстановления x даже в случае знания суммы и вектора весов. В контексте криптографии это может быть интерпретировано как аналог "шумового слоя", повышающего устойчивость к криптоанализу. Тем не менее, анализ научной литературы показывает, что область применения неинъективных рюкзачных векторов остается недостаточно изученной, что открывает перспективное направление для дальнейших исследований и обуславливает научную новизну диссертационной работы.

На основе проведенного анализа и выявленных свойств сюръективных рюкзачных структур разработана модификация классической схемы Меркла–Хеллмана.

Как показано во второй главе, сюръективные рюкзаки образуют особый класс комбинаторных структур, для которых множество достижимых значений целевой функции представляет собой непрерывный диапазон целых чисел. Для каждой такой формы существует хотя бы одно решение при любом допустимом значении правой части уравнения. Было установлено, что задачи, построенные на основе сюръективных рюкзаков, относятся к числу легкорешаемых, поскольку для них реализован алгоритм нахождения всех решений, работающий за линейное время относительно числа переменных и количества решений.

При этом показано, что число сюръективных рюкзаков экспоненциально возрастает с ростом размерности задачи, что делает невозможным атаки методом перебора по пространству ключей при их использовании в качестве закрытых ключей в криптосистеме. Одновременно доказано, что применение сильного модульного умножения к сюръективным рюкзакам нарушает их сюръективность и преобразует задачу в труднорешаемую. В то же время, множество допустимых решений при этом сохраняется, что позволяет контролировать переход между различными уровнями вычислительной сложности без изменения базовой структуры множества решений. Эти свойства делают сюръективные формы

потенциальным кандидатом для использования в качестве закрытых ключей в криптосистеме.

В рамках проведенного исследования на основе установленных свойств разработана модификация схемы Меркла–Хеллмана, использующая сюръективные формы вместо сверхрастущих последовательностей. Такая замена обеспечивает возможность защиты от существующих атак на классические рюкзачные криптосистемы.

В разработанных алгоритмах используются параметры, определенные в разделе 3.1: длина вектора $n = 200$, количество пропущенных начальных элементов $p = 5$, плотность $d^* \approx 1{,}033$ и диапазон значений модуля $M \in [\sum_{i=1}^{n} a_i ; \sum_{i=1}^{n} a_i + a_{n-1}]$. Эти параметры обеспечивают баланс между вычислительной сложностью, структурной полнотой рюкзачных форм и устойчивостью к методам редукции решеток.

На основе указанных принципов разработаны процедуры генерации параметров, построения сюръективного рюкзака, модульного преобразования и восстановления решения, которые в совокупности реализуют адаптированную схему Меркла–Хеллмана, устойчивую к атакам с использованием методов редукции решеток.

Алгоритм 4. Генерация ключей в модифицированной схеме

Входные параметры:

- n — размер ключа;
- k — ожидаемое среднее число решений задачи (используется для контроля плотности рюкзака);
- p — число отбрасываемых начальных элементов.

Выход:

- Закрытый ключ (A, M, u), где:
- $A = [a_1, a_2, \dots, a_n]$ — сюръективный вектор с заданным числом средних решений;
- M — модуль для сильного модульного умножения;
- u — случайное число, взаимно простое с M;

- $B = [b_1, b_2, \dots, b_{n-p}]$ — открытый ключ.

Шаг 1. Формируется начальный фрагмент вектора из p элементов: $A = [1, 2, 4, 8, \dots, 2^{p-1}]$.

Шаг 2. Для каждого индекса i от p до n, происходит следующее:

- Вычисляется нижняя граница допустимого значения следующего элемента $lb := max\left\{A[i], \left\lfloor \frac{2^{i+1}}{k} \right\rfloor + 1\right\}$.
- Вычисляется верхняя граница допустимого значения следующего элемента $ub := \sum_{j=1}^{i} a_j$.
- Новый элемент выбирается случайным образом в интервале $a_{i+1} \in (lb, ub)$ и добавляется в вектор A.

Шаг 3. Модуль M выбирается случайным образом в интервале $M \in (\sum_{i=1}^{n} a_i, \sum_{i=1}^{n} a_i + a_{n-1})$.

Шаг 4. Выбирается взаимно простой с M множитель $u \in (1, m-1)$: $\gcd(u, m) = 1$.

Шаг 5. Для каждого i от 1 до $n - p$, вычисляется $b_i = \left(a_{i+p} \cdot u\right) \bmod m$.

Шаг 6. Вернуть пары ключей: открытый ключ $B = (b_1, b_2, \dots, b_n)$, закрытый ключ (A, M, u).

Алгоритм 5. Шифрование одного блока открытого текста в модифицированной схеме

Входные данные:

- $B = [b_1, b_2, \dots, b_{n'}]$ — открытый ключ;
- $S = [s_1, s_2, \dots, s_{n'-l}]$ – битовая строка, представляющая один блок шифруемого сообщения;
- l — длина контрольной суммы (в битах).

Выход:

- C — зашифрованное сообщение.

Шаг 1. Вычислить SHA-256 хеш строки *S*, взять первые *c* бит бинарного представления этого хеша и приписать эти *c* бит контрольной суммы в начало битовой строки *S*: $S' := SHA256(S) \,||\, S$;

Шаг 2. Вычислить и вернуть сумму $C = \sum_{i=1}^{n'} s_i' b_i$.

Алгоритм 6. Поиск решений задачи о сюръективном рюкзаке

Входные параметры:

- $A = [a_1, a_2, \dots, a_n]$ — рюкзачный вектор,
- b — целевое значение суммы,
- k — текущий индекс,
- $X = [x_1, \dots, x_n]$ — частичное решение.
- p — число отброшенных элементов закрытого ключа

Выход:

- $MX = \{X_1, \dots, X_t\}$ — множество всех бинарных векторов $X_j = [x_{j1}, \dots, x_{jn}]$, таких что $\sum_{i=1}^{n} x_{ji}\, a_i = b$

Шаг 1. Инициализация: если $k = None$, установить $k := n$; если $x = None$, установить $X := [0,0, \dots,0]$.

Шаг 2. Базовый случай: если *k=p*: если *b=0*, вернуть *[X]*, иначе — вернуть пустое множество.

Шаг 3. Рекурсивный случай: рассматриваются три возможные ветви в зависимости от соотношения между текущим значением правой части b и частичной суммой элементов рюкзака:

- ***Ветвь 1:*** если $\sum_{i=1}^{k-1} a_i < b$, то элемент a_k обязательно включается: присваиваем $x_k := 1$, вызываем рекурсию с параметрами $(A,\ b - a_k,\ k - 1,\ x, p)$
- ***Ветвь 2:*** если $a_k > b$, то элемент a_k не может быть включен: присваиваем $x_k := 0$, вызываем рекурсию с параметрами $(A,\ b,\ k - 1,\ x, p)$.
- ***Ветвь 3:*** если ни одно из условий выше не выполняется, рассматриваются оба случая: присваиваем $x_k := 1$, вызываем рекурсию с параметрами

$(A,\ b - a_k,\ k - 1,\ x, p)$ и присваиваем $x_k := 0$, вызываем рекурсию с параметрами $(A,\ b,\ k - 1,\ x, p)$.

Шаг 4. Все возвращенные решения из рекурсивных вызовов собираются и возвращаются в виде списка.

Алгоритм 7. Расшифрование одного блока сообщения в модифицированной схеме

Входные параметры:

- C — вектор зашифрованных значений;
- $A = [a_1, a_2, \dots, a_n]$ — закрытый ключ;
- M — модуль, использованный при формировании открытого ключа;
- u — мультипликативный коэффициент для умножения по модулю m;
- p — количество отброшенных элементов в начале вектора;
- l — длина контрольной суммы.

Выход:

- $X = [x_1, \dots, x_n]$ — восстановленное бинарное сообщение длины $n - p$, если существует единственное решение с контрольной суммой.

Шаг 1. Вычисляется обратный элемент $t = u^{-1} mod\ M$.

Шаг 2. Для шифртекста C выполняется обратное преобразование $D = C \cdot t\ mod\ M$.

Шаг 3. Для каждого D вызывается алгоритм полного перебора решений, получаем $MX = \{X_1, \dots, X_t\}$ — множество всех бинарных векторов $X_j = [x_{j1}, \dots, x_{jn}]$, таких что $\sum_{i=1}^{n} x_{ji}\, a_i = C$.

Шаг 4. Для каждого полученного бинарного решения X:

1. Отбрасываются первые p битов $X := X[p + 1:]$
2. Выполняется разделение результата:
   - $cs := X[: l + 1]$ – контрольная сумма
   - $msg :=\ X[l + 1:]$ — сообщение-кандидат.
3. Вычисляется SHA-256 хеш строки msg, берутся первые l бит бинарного представления этого хеша: $cs2 = SHA256(msg)[: l + 1]$

4. Если $cs2 = cs$, сообщение msg добавляется в список корректных решений.

Шаг 5. Если найдено единственное корректное решение, оно возвращается, иначе — выдается предупреждение.

3.3 Экспериментальное исследование разработанных алгоритмов

Для проверки корректности предложенных конструкций, оценки их вычислительных характеристик и подтверждения применимости разработанных методов в практических задачах был проведен комплекс экспериментальных исследований. Целью экспериментов являлась в первую очередь верификация теоретических положений — проверка свойств сюръективных рюкзачных форм, устойчивости их структуры при модульных преобразованиях и воспроизводимости аналитических зависимостей между параметрами задачи и ее вычислительной сложностью.

Исследования выполнялись на программно-аппаратном стенде, реализованном на персональном компьютере со следующими характеристиками:

- Процессор: Intel Core i7-1165G7, 4 ядра, 2.80 ГГц
- Оперативная память: 16 ГБ DDR4
- Накопитель: SSD NVMe 512 ГБ
- Операционная система: Ubuntu 22.04 LTS, ядро 5.15

Программный комплекс был реализован на языке Python 3.11 [68] с использованием модулей NumPy [69] и hashlib [70]. В состав комплекса входят следующие модули:

- KeyGenModule — генерация исходных параметров и построение сюръективных рюкзачных форм с последующим модульным преобразованием;
- Encryptor — модуль шифрования сообщений с использованием открытого ключа и встроенного контроля целостности;
- Decryptor — модуль расшифрования, включающий обратное преобразование, поиск решений задачи рюкзака и проверку контрольной суммы;

- LLLTester — модуль для имитации атак на основе редукции решеток (алгоритм LLL);
- PerfProfiler — модуль автоматического замера времени выполнения ключевых операций.

Такое построение позволяет моделировать полный цикл функционирования исследуемых алгоритмов — от генерации исходных данных до анализа полученных результатов. При этом криптографическое применение рассматривается как один из частных примеров использования разработанных структур, демонстрирующий их корректность в условиях практических вычислений. Основное внимание уделялось оценке алгоритмической состоятельности и подтверждению аналитических выводов о влиянии параметров рюкзачных форм на вычислительную сложность.

1. Проверка корректности преобразований

Одним из ключевых требований к разработанным алгоритмическим структурам является их корректность и воспроизводимость, то есть гарантированное восстановление исходных данных после выполнения прямого и обратного модульных преобразований при фиксированных параметрах задачи. Это свойство необходимо как для практического применения алгоритмов в задачах обработки информации, так и для подтверждения внутренней согласованности предложенной теоретической модели.

Цель эксперимента

Целью данного этапа тестирования являлась проверка корректности работы базовых процедур преобразования рюкзачных форм — генерации, модульного преобразования и обратного восстановления. Под корректной работой алгоритма понимается выполнение условия взаимно-однозначного соответствия между исходными и восстановленными данными, то есть для каждого случайно сгенерированного входного вектора X после последовательного применения прямого преобразования $Y = A(X)$ и обратного $X' = B(Y)$ выполняется равенство $X = X'$.

Исходные условия эксперимента

В эксперименте использовались следующие значения параметров:

- Размер блока: *n=200;*
- Размер контрольной суммы: *l = 16;*
- Параметр плотности рюкзака: $d \approx 1{,}031$;
- Среднее число допустимых решений: $\widetilde{N} \approx 2000$;
- Число отбрасываемых начальных элементов: *p = 5;*

Подбор указанных параметров основан на предварительном анализе, представленном в разделе 3.1, и соответствует области, где сюръективные формы демонстрируют наибольшую плотность при умеренном числе допустимых решений.

Методика проведения теста

Эксперимент проводился в автоматическом режиме с использованием разработанного программного комплекса, описанного в данном разделе. Общая схема тестирования включала следующие этапы:

1. Генерация наборов сюръективных рюкзачных векторов с заданными характеристиками n и $\widetilde{N}$ и их модульное преобразование;
2. Формирование случайных бинарных последовательностей длиной от 200 до 1000 бит, моделирующих исходные данные;
3. Разбиение каждой последовательности на блоки длиной $n - l - p$ (в рассматриваемом случае 179 бит) и добавление к каждому блоку контрольной суммы длиной $l = 16$ бит.
4. Выполнение прямого преобразования для каждого блока с использованием преобразованных по модулю сюръективных рюкзачных векторов.
5. Применение обратного преобразования, включающего модульное преобразование, поиск всех допустимых решений и отбор корректного блока по контрольному параметру.
6. Сравнение исходных и восстановленных данных для проверки точности преобразований.

Для повышения достоверности результаты собирались по 100000 независимым запускам при фиксированных параметрах. Во всех случаях восстановленные последовательности полностью совпадали с исходными. Это подтверждает корректность реализованных процедур преобразования и обратного восстановления для выбранных параметров n, l, p и d, а также воспроизводимость вычислительных результатов.

Условия и результаты тестирования приведены в таблице 1.

Таблица 1. Результаты тестирования корректности расшифрования

Параметр	**Значение**
Количество тестов	100 000
Длина сообщений	до 1 000 бит
Размер блока	200 бит
Размер контрольной суммы	16 бит
Количество ошибок	0
Доля корректных преобразований	100%

Несмотря на полученные результаты, стоит отметить, что при больших объемах преобразуемой информации теоретически возможны коллизии, приводящие к некорректному обратному преобразованию. Для таких ситуаций в практической реализации может быть предусмотрен механизм повторного прямого преобразования при множественности допустимых решений, либо других признаках искажения.

2. Оценка производительности

Для оценки практической применимости разработанных алгоритмов преобразования рюкзачных структур необходимо исследовать их вычислительные характеристики: время генерации исходных параметров, выполнение прямого и обратного преобразований, а также устойчивость временных характеристик при масштабировании размерности задачи. Эти параметры являются ключевыми для анализа пригодности предложенных методов при работе с большими объемами данных.

Среда тестирования

Для базового сравнения использовалась реализация алгоритма RSA [71] из пакета cryptography.hazmat.primitives.asymmetric.rsa [72], основанная на библиотеке OpenSSL [73]. Она рассматривалась в качестве эталонной модели, демонстрирующей типичный уровень производительности алгоритмов криптографического преобразования данных, работающих с большими целыми числами и линейными преобразованиями.

В ходе тестирования производились замеры времени выполнения следующих операций:

- генерация параметров и исходных последовательностей (аналог генерации ключей);
- выполнение прямого преобразования бинарного блока (аналог шифрования);
- выполнение обратного преобразования и проверки корректности (аналог расшифрования).

Каждое измерение повторялось 10 000 раз, после чего усреднялось. Замеры выполнялись как для предлагаемой рюкзачной криптосистемы, так и для алгоритма RSA с ключом 2048 бит. Результаты тестирования представлены в таблице 2.

Таблица 2. Сводная таблица результатов

Операция	**RSA (2048 бит)**	**Схема на основе рюкзачных форм**
Генерация ключей	340 мс	0,05 мс
Шифрование	0,6 мс	0,6 мс
Расшифрование	1,7 мс	26 мс

Как видно из таблицы, время прямого преобразования в разработанных алгоритмах сопоставимо с эталонным RSA — порядка 0,6 мс. Что объясняется тем, что операция сводится к последовательному перемножению и суммированию элементов рюкзачного вектора, что реализуется с высокой эффективностью. Это делает алгоритм подходящим для использования в условиях ограниченного времени отклика.

Формирование исходных параметров выполняется существенно быстрее, чем генерация ключей в RSA, поскольку процесс не требует трудоемких операций с большими простыми числами или факторизации. Генерация параметров сводится к построению псевдослучайной последовательности и последующему преобразованию по заданным формулам, что в среднем занимает около 0,05 мс.

Наибольшие вычислительные затраты приходятся на этап обратного преобразования, среднее время выполнения которого составило около 26 мс. Этот результат объясняется особенностями алгоритма:

- необходимостью перебора допустимых решений задачи;
- выполнением процедур проверки по контрольным суммам, обеспечивающим корректность восстановления исходных данных;
- использованием линейного поиска и фильтрации кандидатов, часть которых формируется итеративно.

Следует отметить, что текущая реализация выполнена полностью на языке Python без применения JIT-компиляции, низкоуровневых модулей или параллельных вычислений [74]. При переносе алгоритмов на компилируемые языки (C/C++) [75], либо при использовании оптимизирующих библиотек numba [76] или cython [77], ожидается значительное снижение времени выполнения.

Даже при существующих параметрах время обратного преобразования остается в пределах, приемлемых для практического применения, особенно в задачах, где подобные операции выполняются значительно реже, чем прямые преобразования.

Кроме того, в ходе экспериментов были выявлены следующие особенности алгоритма:

- Стабильность результатов. Разброс времени выполнения между запусками не превышал 2–5% от среднего значения, что указывает на устойчивость реализации и предсказуемость временных характеристик.
- Масштабируемость. При увеличении размерности входных данных наблюдается линейный рост времени выполнения, что подтверждает

полиномиальную зависимость вычислительной сложности от длины входа.

- Потенциал параллелизации. Поиск решений реализован рекурсивно и может быть распределен между потоками или вычислительными узлами, что открывает возможность ускорения при использовании многопроцессорных архитектур.

3. Стойкость к LLL-атакам

Одним из ключевых этапов экспериментального исследования стало изучение устойчивости разработанных структур и алгоритмов к методам редукции решеток, которые представляют собой один из наиболее эффективных инструментов для поиска приближенных решений целочисленных задач оптимизации.

Особое внимание было уделено алгоритму Ленстры–Ленстры–Ловаса (LLL), который выполняет редукцию базиса решетки, формируя систему ортогональных коротких векторов и, как следствие, восстанавливая исходное решение. Устойчивость к LLL-атакам является критически важной для рюкзачных криптосистем, поскольку подобные атаки являются одними из наиболее эффективных средств криптоанализа для данной группы алгоритмов.

В рамках эксперимента моделировалось воздействие метода редукции решеток на разработанные структуры, включая сюръективные рюкзачные векторы и векторы, полученные их них с помощью сильного модульного умножения. Целью проверки являлось определение устойчивости задачи к восстановлению исходных решений после применения LLL-преобразования, а также выявление зависимости доли успешных восстановлений от размерности и плотности вектора.

Экспериментальная процедура включала следующие этапы:

1. Генерация исходного и преобразованного рюкзачных векторов в соответствии с выбранными параметрами n, M, l, p и плотностью d.
2. Генерация случайных решений и вычисление целевых значений на их основе.

3. Построение решетки по преобразованному по модулю сюръективному вектору и решению.
4. Применение алгоритма LLL к решетке и анализ полученных базисных векторов для восстановления решений исходной задачи.

Полученные результаты сведены в таблицу 3:

Таблица 3. Зависимость доли успешных решений задачи алгоритмом LLL от длины блока исходного решения

Длина блока (бит)	**Доля успешных решений (%)**
8	0,47
12	0,33
16	0,18
32	0,05
64	0,008
128	0,0003

В таблице под длиной блока понимается не размер исходного решения, а число подряд идущих бит, корректно восстановленных в результате процедуры решения задачи после редукции решетки. Поскольку каждый эксперимент проводился при фиксированной длине рюкзачного вектора, величина блока отражает не исходные параметры задачи, а глубину совпадения между найденным решением и эталонным вектором.

Для экземпляров с плотностью, превышающей единицу, алгоритм редукции базиса LLL часто возвращает допустимый, но не идентичный исходному, вектор. Такие решения обычно воспроизводят часть исходной структуры задачи, особенно в старших разрядах, где влияние крупных коэффициентов более выражено. Это позволяет оценивать не только факт успешного нахождения решения, но и степень приближения, характеризующую поведение алгоритма в пространстве близких к оптимуму векторов. В связи с этим была введена метрика — доля экспериментов, в которых восстановленный вектор совпадает как минимум в N первых битах с эталонным решением. При

варьировании N от 8 до 128 исследовалось, как изменяется вероятность частичного совпадения решений.

Полученные результаты показали, что с увеличением требуемой длины совпадения вероятность успешного восстановления быстро уменьшается. Каждый дополнительный бит, совпадающий с эталоном, требует все более точного приближения, что при высоких плотностях становится вычислительно труднодостижимым.

Таким образом, проведенный анализ подтверждает, что даже при частичном совпадении решений редукция решетки не дает существенного приближения к исходному вектору, что указывает на высокую устойчивость предложенных методов к решениям с помощью алгоритмов редукции базиса решетки.

Выводы к главе 3

Рассмотрена реализация модифицированных рюкзачных структур на основе сюръективных форм и проведен подбор параметров, обеспечивающих полиномиальную разрешимость при высокой плотности и ограниченных вычислительных затратах на решение.

Основное внимание уделено влиянию длины вектора, плотности, диапазона коэффициентов и числа допустимых решений на сложность поиска решений. Показано, что при $n = 200$ и плотности $d \approx 1{,}03$ экземпляры становятся устойчивыми к методам редукции решеток, включая алгоритм Ленстры–Ленстры–Ловаса (LLL), при сохранении возможности их эффективного решения предложенным в главе 2 алгоритмом.

Для контроля числа решений использован метод регулирования диапазона коэффициентов, а однозначность восстановления решений в вычислительных экспериментах обеспечена с помощью контрольных сумм, формируемых усеченными хэш-функциями.

Выбор модуля M обоснован с точки зрения поддержания требуемой плотности и сохранения конфигурации допустимых решений.

Проведено экспериментальное исследование реализованных алгоритмов, направленное на проверку корректности и производительности предложенного алгоритма решения и устойчивости экземпляров к методам редукции решеток.

На выборке из 100 000 случайных входных данных длиной до 1000 бит при параметрах $n = 200, l = 16, p = 5, \widetilde{N} = 2000, d^* \approx 1{,}031$ во всех случаях получено корректное восстановление исходных данных, что подтверждает точность и устойчивость предложенного подхода. Коллизии и неоднозначные решения не наблюдались, что свидетельствует о согласованности аналитических и вычислительных результатов.

Показана применимость реализованных методов в прикладных задачах защиты информации. Предложена реализация асимметричной схемы шифрования с использованием сюръективных рюкзачных векторов. Производительность разработанных процедур сравнивалась с базовой реализацией алгоритма RSA. Среднее время генерации параметров составило 0,05 мс, вычисления комбинаций — около 0,6 мс, а восстановления решений — 26 мс. Несмотря на большее время последнего этапа, реализация выполнена на интерпретируемом языке Python, поэтому в компилируемом исполнении (например, на C/C++) ожидается значительное ускорение, особенно при использовании параллельных вычислений.

Отдельное внимание уделено устойчивости к алгоритмам решения с использованием методов редукции базиса решетки. Моделирование LLL-редукции в среде SageMath подтвердило, что при плотности $d \approx 1{,}03$ вероятность успешного восстановления решения экспоненциально уменьшается с ростом длины совпадения. Так, при 8 совпадающих битах вероятность успешного восстановления составила около 0,47 %, а при 128 битах — менее 0,0003 %.

Это указывает на устойчивость экземпляров высокой плотности к частичному восстановлению решений и подтверждает применимость разработанных алгоритмов в задачах защиты информации.

Заключение

В работе проведено комплексное исследование комбинаторных и вычислительных свойств задач рюкзачного типа, направленное на выявление зависимостей между параметрами задачи, структурой множества ее решений и эффективностью применяемых алгоритмов. Полученные результаты образуют методическую основу для построения и анализа классов задач с заранее заданными характеристиками трудности решения.

1. Теоретически обоснованы и получены аналитические формулы, позволяющие вычислять среднее число допустимых решений задачи об ограниченном рюкзаке для всех экземпляров фиксированной размерности. Выведены выражения для среднего значения целевого функционала через количество решений подзадач меньшей размерности. Эти результаты обеспечивают возможность аналитической оценки множества решений и служат основой для последующего анализа вычислительной сложности.
2. Выделен и исследован класс сюръективных линейных форм, для которых множество достижимых значений представляет собой непрерывный диапазон целых чисел. Установлены необходимые и достаточные условия сюръективности и найдены параметры, влияющие на сложность решения таких задач. Разработан алгоритм полного перебора решений сюръективных форм, обладающий линейной сложностью по числу переменных и количеству решений. Показано, что сюръективные формы образуют структурно регулярные классы, для которых возможно построение эффективных алгоритмов поиска решений при сохранении высокой асимптотической сложности для других методов.
3. Исследованы линейные формы с разрывами в области значений и построен метод их формирования с контролируемым числом разрывов. Установлена зависимость числа и расположения разрывов от структуры коэффициентов и показано, что постепенное увеличение числа разрывов

моделирует переход от сюръективных форм к нерегулярным экземплярам, сопровождающийся ростом вычислительной сложности.

4. На основе аналитических выражений разработаны методы подбора параметров задачи, обеспечивающих баланс между вычислительной эффективностью и устойчивостью экземпляров к методам редукции решеток. Показано, что выбор размерности, плотности и коэффициентов рюкзачного вектора позволяет конструировать классы задач, труднорешаемые в стандартных подходах, но разрешимые с использованием предложенного алгоритма. Установленные зависимости позволяют классифицировать экземпляры по степени трудности решения и выделять области параметров, устойчивые к методам редукции решеток.
5. Разработаны методы параметрических преобразований экземпляров, позволяющие изменять сложностью решения задачи при сохранении структуры множества решений. Показано, что такие преобразования обеспечивают переход от легкорешаемых к труднорешаемым экземплярам при контролируемых изменениях плотности. Полученные методы могут применяться в задачах кодирования, защиты и хранения информации, где требуется контролируемая вычислительная сложность.
6. Создан и реализован программный комплекс, включающий модули генерации параметров, построения форм, вычисления решений и анализа их структуры. Проведенные эксперименты подтвердили корректность аналитических моделей и адекватность предложенных алгоритмов. Разработанное программное обеспечение обеспечивает воспроизводимость вычислений и может использоваться в дальнейшем для генерации и анализа классов задач комбинаторной оптимизации.

Список использованной литературы

1. Волков М. С. А. Комбинаторные свойства задачи об ограниченном рюкзаке // Прикладная дискретная математика. — 2024. — № 63. — С. 117–130.
2. Волков М. С. А., Гордеев Э. Н., Леонтьев В. К. О среднем числе допустимых решений в задаче о рюкзаке // Прикладная дискретная математика. — 2025. — № 68. — С. 103–113.
3. Волков М. С. А., Гордеев Э. Н. Применение неинъективных векторов в ранцевых криптосистемах // Безопасность информационных технологий. — 2025. — Т. 32, № 1. — С. 122–131.
4. Волков М. С. А. Анализ и реализация криптосистемы на основе неинъективных ранцев // Безопасность информационных технологий. — 2025. — Т. 32, № 2. — С. 100–111.
5. Леонтьев В. К., Гордеев Э. Н., Волков М. С. А. Классическая непрерывность и ее дискретный вариант // Прикладная физика и математика. — 2022. — № 1. — С. 31–37.
6. Волков М. С. А., Гордеев Э. Н. О непрерывности линейных форм // Безопасные информационные технологии: материалы XII Междунар. науч.-техн. конф., посвященной 25-летию кафедры ИУ8, Москва, 02 ноября 2023 г. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2024. — С. 16–20.
7. Волков, М. С. А., Гордеев Э. Н., Леонтьев В. К. О свойствах решений обобщенной задачи о рюкзаке // Безопасные информационные технологии: материалы XII Междунар. науч.-техн. конф., Москва, 02 ноября 2023 г. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2024.
8. Волков М. С. А., Гордеев Э. Н. Исследование и применение сюръективных рюкзаков в криптографии // Безопасные информационные технологии: материалы XIII Междунар. науч.-техн. конф., Москва, 01 ноября 2024 г. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2024. — С. 54–56.

9. Volkov M. S. A. Application of the Method of Coefficients for the Analysis of Combinatorial Properties of the Knapsack Problem // Science, Engineering and Business: Collection of materials V Interacademic Conference for Graduate Students and Young Researchers, Moscow, April 18–19, 2023. — Moscow: Bauman MSTU, 2023. — P. 303–308.
10. Garey M. R., Johnson D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. — San Francisco: W. H. Freeman, 1979.
11. Papadimitriou C. H. Computational Complexity. — Reading, MA: Addison-Wesley, 1994.
12. Cook S. A. The complexity of theorem-proving procedures // Proceedings of the 3rd Annual ACM Symposium on Theory of Computing. — 1971. — P. 151–158.
13. Levin L. A. Universal sequential search problems // Problems of Information Transmission. — 1973. — Vol. 9. — P. 265–266.
14. Karp R. M. Reducibility among combinatorial problems // Complexity of Computer Computations. — 1972. — P. 85–103.
15. Arora S., Barak B. Computational Complexity: A Modern Approach. — Cambridge: Cambridge University Press, 2009.
16. Kellerer H., Pferschy U., Pisinger D. Knapsack Problems. — Berlin: Springer, 2004. — 548 p. DOI: 10.1007/978-3-540-24777-7.
17. Schaefer T. J. The complexity of satisfiability problems // Proceedings of the Tenth Annual ACM Symposium on Theory of Computing (STOC ’78). — 1978. — P. 216–226.
18. Pisinger D. Algorithms for Knapsack Problems. — PhD Thesis. — University of Copenhagen, 1999.
19. Lagarias J. C. Knapsack public key cryptosystems and Diophantine approximation // Advances in Cryptology: Proceedings of Crypto 83. — Boston, MA: Springer US, 1984. — P. 3–23.
20. Pisinger D. Where are the hard knapsack problems? // Computers & Operations Research. — 2005. — Vol. 32, No. 9. — P. 2271–2284.

21. Yannakakis M. Expressing combinatorial optimization problems by linear programs // Journal of Computer and System Sciences. — 1991. — Vol. 43, No. 3. — P. 441–466.
22. Odlyzko A. M. The rise and fall of knapsack cryptosystems // Cryptology and Computational Number Theory. — Providence, RI: American Mathematical Society, 1990. — P. 75–88.
23. Ausiello G., Crescenzi P., Gambosi G., Kann V., Marchetti-Spaccamela A., Protasi M. Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties. — Berlin: Springer, 1999.
24. Martello S., Toth P. Knapsack Problems: Algorithms and Computer Implementations. — New York: John Wiley & Sons, 1990.
25. Bellman R. Dynamic programming // Science. — 1966. — Vol. 153, No. 3731. — P. 34–37.
26. Pferschy U. Dynamic programming revisited: improving knapsack algorithms // Computing. — 1999. — Vol. 63, No. 4. — P. 419–430.
27. Kolesar P. J. A branch and bound algorithm for the knapsack problem // Management Science. — 1967. — Vol. 13, No. 9. — P. 723–735.
28. Abidin S. Greedy approach for optimizing 0–1 knapsack problem // Communications on Applied Electronics. — 2017. — Vol. 7, No. 6. — P. 1–3.
29. Ibarra O. H., Kim C. E. Fast approximation algorithms for the knapsack and sum of subset problems // Journal of the ACM. — 1975. — Vol. 22, No. 4. — P. 463–468.
30. Chu P. C., Beasley J. E. A genetic algorithm for the multidimensional knapsack problem // Journal of Heuristics. — 1998. — Vol. 4, No. 1. — P. 63–86.
31. Brickell E. F. Solving low density knapsacks // Advances in Cryptology: Proceedings of Crypto ’83. — Boston, MA: Springer US, 1984. — P. 25–37.
32. Shamir A. On the cryptocomplexity of knapsack systems // Proceedings of the 11th Annual ACM Symposium on Theory of Computing. — 1979. — P. 118–129.
33. Колпаков Р. М., Посыпкин М. А. Верхняя и нижняя оценки трудоемкости метода ветвей и границ для задачи о ранце // Дискретная математика. — 2010. — Т. 22, № 1. — С. 58–73.

34. Колпаков Р. М., Посыпкин М. А., Си Ту Тант Син. Верхняя оценка сложности одного из вариантов метода ветвей и границ для задачи о сумме подмножеств // International Journal of Open Information Technologies. — 2016. — Т. 4, № 2. — С. 1–6.
35. Колпаков Р. М., Посыпкин М. А., Сигал И. Х. О нижней оценке вычислительной сложности одной параллельной реализации метода ветвей и границ // Автоматика и телемеханика. — 2010. — № 10. — С. 156–166.
36. Колпаков Р. М. Оптимальная стратегия решения частного случая задачи о ранце методом ветвей и границ // Вестник Московского университета. Серия 1. Математика. Механика. — 2021. — № 3. — С. 13–22.
37. Smith-Miles K., Lopes L. Measuring instance difficulty for combinatorial optimization problems // Computers & Operations Research. — 2012. — Vol. 39, No. 5. — P. 875–889.
38. Wilf H. S. Generatingfunctionology. — 3rd ed. — Natick, MA: A K Peters/CRC Press, 2006.
39. Егорычев Г. П. Интегральное представление и вычисление комбинаторных сумм. — Новосибирск: Наука, 1977. — 285 с.
40. Леонтьев В. К. Комбинаторика и информация. Часть 1. Комбинаторный анализ: учебное пособие. — М.: МФТИ, 2015. — 174 с.
41. Riedel M., Mahmoud H. Egorychev method: a hidden treasure // La Matematica. — 2023. — Vol. 2, No. 4. — P. 893–933.
42. Frieze A. M. On the Lagarias–Odlyzko algorithm for the subset sum problem // SIAM Journal on Computing. — 1984. — Vol. 13, No. 2. — P. 387–391.
43. Nguyen P. Q., Stern J. Merits and limits of the LLL reduction for lattice-based cryptanalysis // Advances in Cryptology — CRYPTO ’97. LNCS, vol. 1294. — Berlin: Springer, 1997. — P. 395–403.
44. Lenstra A. K., Lenstra H. W., Lovász L. Factoring polynomials with rational coefficients // Mathematische Annalen. — 1982. — Vol. 261. — P. 515–534. DOI: 10.1007/BF01457454.

45. Lagarias J. C., Odlyzko A. M. Solving low-density subset sum problems // Journal of the ACM. — 1985. — Vol. 32. — P. 229–246.
46. Coster M. J., Joux A., LaMacchia B. A., Odlyzko A. M., Schnorr C. P., Stern J. Improved low-density subset sum algorithms // Computational Complexity. — 1992. — Vol. 2. — P. 111–128.
47. Ajtai M., Kumar R., Sivakumar D. A sieve algorithm for the shortest lattice vector problem // Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC ’01). — New York: ACM, 2001. — P. 601–610. DOI: 10.1145/380752.380857.
48. Joux A., Stern J. Lattice reduction: A toolbox for the cryptanalyst // Journal of Cryptology. — 1998. — Vol. 11, No. 3. — P. 161–185.
49. Van Hoeij M. Factoring polynomials and the knapsack problem // Journal of Number Theory. — 2002. — Vol. 95, No. 2. — P. 167–189.
50. Гордеев Э. Н., Леонтьев В. К. Производящие функции в задаче о рюкзаке // Доклады Академии наук. — 2018. — Т. 481, № 5. — С. 478–480. DOI: 10.31857/s086956520002139-5.
51. Гордеев Э. Н., Леонтьев В. К. О некоторых комбинаторных свойствах задачи о рюкзаке // Журнал вычислительной математики и математической физики. — 2019. — Т. 59, № 8. — С. 1439–1447. DOI: 10.1134/S0044466919080076.
52. Леонтьев В. К., Гордеев Э. Н. Зависимость среднего числа решений в задаче о рюкзаке от параметров области ограничений // Безопасные информационные технологии: труды XI Междунар. науч.-техн. конф. — М.: МГТУ им. Н. Э. Баумана, 2021. — С. 85–90.
53. Кнут Д. Э. Искусство программирования. Том 1: Основные алгоритмы. — 3-е изд. — М.: Вильямс, 2002. — 720 с.
54. Гордеев Э. Н., Леонтьев В. К. О некоторых комбинаторных свойствах задачи о ранце // Журнал вычислительной математики и математической физики. — 2019. — Т. 59, № 8. — С. 1439–1447. DOI: 10.1134/S0044466919080076.
55. Саломаа А. Криптография с открытым ключом. — М.: Мир, 1995. — 318 с.

56. Merkle R., Hellman M. Hiding information and signatures in trapdoor knapsacks // IEEE Transactions on Information Theory. — 1978. — Vol. 24, No. 5. — P. 525–530.
57. Khalaf R. Z., Hamza B. H., Thakwan A. J. Attacking the Knapsack Public-key Cryptosystem // Webology. — 2022. — Vol. 19, No. 1. — P. 5302–5309. DOI: 10.14704/web/v19i1/web19356.
58. Liu J., Bi J., Xu S. An improved attack on the basic Merkle–Hellman knapsack cryptosystem // IEEE Access. — 2019. — Vol. 7. — P. 59388–59393. DOI: 10.1109/ACCESS.2019.2913678.
59. Koskinen A. Non-Injective Knapsack Public-Key Cryptosystems // Proceedings of the 3rd Central European Conference on Cryptography (CECC). — 2003.
60. Schnorr C. P., Euchner M. Lattice basis reduction: improved practical algorithms and solving subset sum problems // Mathematical Programming. — 1994. — Vol. 66. — P. 181–199.
61. Rueppel R. A. The knapsack as a nonlinear function // Analysis and Design of Stream Ciphers. — Berlin: Springer, 1986. — P. 163–191.
62. Thangavel M., Varalakshmi P. A novel public key cryptosystem based on Merkle–Hellman Knapsack Cryptosystem // 2016 Eighth International Conference on Advanced Computing (ICoAC). — IEEE, 2017. — P. 117–122.
63. Federal Information Processing Standards Publication 180-4. Secure Hash Standard (SHS). URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf (дата обращения: 15.02.2025).
64. Zhang H., Han W., Lai X., Lin D., Ma J., Li J. Survey on cyberspace security // Science China Information Sciences. — 2015. — Vol. 58, No. 11. — P. 1–43. DOI: 10.1007/s11432-015-5433-4.
65. Lyubashevsky V., Palacio A., Segev G. Public-key cryptographic primitives provably as secure as subset sum // Lecture Notes in Computer Science. — 2010. — P. 382–400. DOI: 10.1007/978-3-642-11799-2_23.

66. Shamir A. A polynomial-time algorithm for breaking the basic Merkle–Hellman cryptosystem // 23rd Annual Symposium on Foundations of Computer Science (FOCS). — 1982.
67. Furst M., Kannan R. Succinct certificates for almost all subset sum problems // SIAM Journal on Computing. — 1989. — Vol. 18, No. 3. — P. 550–558.
68. Python 3.11.0 Documentation [Электронный ресурс]. URL: https://docs.python.org/3/ (дата обращения: 15.10.2024).
69. NumPy v1.25 Reference Guide [Электронный ресурс]. URL: https://numpy.org/doc/stable/ (дата обращения: 15.10.2024).
70. hashlib — Secure hashes and message digests [Электронный ресурс]. URL: https://docs.python.org/3/library/hashlib.html (дата обращения: 15.10.2024).
71. Rivest R. L., Shamir A., Adleman L. M. A method for obtaining digital signatures and public-key cryptosystems // Communications of the ACM. — 1978. — Vol. 21, No. 2. — P. 120–126.
72. Cryptography.io: RSA [Электронный ресурс]. URL: https://cryptography.io/en/latest/hazmat/primitives/asymmetric/rsa/ (дата обращения: 15.10.2024).
73. OpenSSL: Cryptography and SSL/TLS Toolkit [Электронный ресурс]. URL: https://www.openssl.org/ (дата обращения: 15.10.2024).
74. Beazley D. Python Essential Reference. — Addison-Wesley, 2009.
75. Sutter H., Alexandrescu A. C++ Coding Standards: 101 Rules, Guidelines, and Best Practices. — Addison-Wesley, 2004.
76. Numba: A High Performance Python Compiler [Электронный ресурс]. URL: https://numba.pydata.org/ (дата обращения: 15.10.2024).
77. Cython: C-Extensions for Python [Электронный ресурс]. URL: https://cython.org/ (дата обращения: 15.10.2024).